

RESEARCH

Open Access



Multi-strategy enhanced marine predator algorithm: performance investigation and application in intrusion detection

Zhongmin Wang¹, Yujun Zhang², Jun Yu³, YuanYuan Gao⁴, Guangwei Zhao^{5,6}, Essam H. Houssein⁷ and Rui Zhong^{8*}

*Correspondence:
zhongrui@iic.hokudai.ac.jp

¹ College of Tropical Crops, Yunnan Agricultural University, Pu'er, Yunnan, China

² School of New Energy, Jingchu University of Technology, Jingmen, China

³ Institute of Science and Technology, Niigata University, Niigata, Japan

⁴ School of Big Data and Computer Science, Shanxi Institute of Science and Technology, Jincheng, Shanxi, China

⁵ College of Computer and Artificial Intelligence, Huanghuai University, Zhumadian, China

⁶ Henan Key Laboratory of Smart Lighting, Zhumadian, China

⁷ Faculty of Computers and Information, Minia University, Minia, Egypt

⁸ Information Initiative Center, Hokkaido University, Sapporo, Japan

Abstract

Marine Predator Algorithm (MPA) is a recently proposed population-based metaheuristic algorithm (MA), and its effectiveness has been proven in many stochastic optimization challenges. However, like most MAs, MPA suffers from shortcomings such as imbalanced search preferences and stagnation in the late phase of optimization. Therefore, this paper presents a Multi-strategy Enhanced Marine predator algorithm (MEMPA), where (1) a low-discrepancy Sobol sequence is introduced to generate promising initial solutions in the high-dimensional search domain, (2) the mutualism mechanism is integrated to enhance the ability to escape from local optima, and (3) a distance-based selection scheme is embedded to enhance the diversity of the population during optimization. We conducted comprehensive numerical experiments and rigorous statistical analysis to evaluate the performance of MEMPA against eleven well-known MAs in the CEC2020 benchmark and six classic engineering problems. The experimental results and statistical analysis confirm the efficiency and effectiveness of our proposed MEMPA comprehensively. Finally, we extend the proposed MEMPA to optimize the hyper-parameters of the Extreme Learning Machine (ELM) for intrusion detection, and the performance of MEMPA-ELM increases the average accuracy by 0.79% than the second-best model. The source code of this research can be downloaded at <https://github.com/RuiZhong961230/MEMPA>.

Keywords: Metaheuristic algorithm (MA), Marine predator algorithm (MPA), Multi-strategy enhancement, Engineering optimization, Intrusion detection

Introduction

In the past decade, a large number of novel metaheuristic algorithms (MAs) have been introduced due to their high efficiency in handling complex optimization problems [1, 2]. Since the birth of MAs, at least about 350 algorithms inspired by hundreds of animals, plants, microorganisms, natural phenomena, and human social phenomena have been widely proposed [3, 4]. Compared with traditional optimization methods, such as exhaustive search, gradient descent, and Newton's method, the solving time of these methods will increase exponentially with the increase of problem size, and it is difficult to find satisfactory solutions, but MAs can often get satisfactory results within an

affordable computational budget. Therefore, MA has become one of the most popular optimization techniques in both academic research and real-world applications [5–7].

At present, a common approach to classifying the inspiration of MAs is presented in Fig. 1. Four categories are contained in this classification diagram: physics/chemistry/math-inspired, bio-inspired, human-inspired, and swarm intelligence approaches. Numerous advanced intelligent optimization algorithms have both theoretically and empirically demonstrated that no single MA is universally effective for solving all optimization problems. This principle, known as the No Free Lunch (NFL) Theorem [8], asserts that while a specific MA may excel on one set of problems, it may perform poorly on others. This inherent limitation ensures that the field of optimization remains highly dynamic, with ongoing research aimed at developing new MAs and enhancing existing methods to address different problem domains. As a result, the introduction of innovative approaches and algorithmic improvements that push the boundaries of performance and applicability is popular in the research domain [9–12]. The NFL Theorem thus plays a pivotal role in driving the continuous evolution of optimization techniques, keeping this area of research vibrant and developing.

Motivated by the Brownian motion and foraging behaviors of marine predators, Faramarzi et al. [13] proposed a new population-based MA called Marine Predator Algorithm (MPA). Compared to other MAs, MPA offers several key advantages, including a minimal number of parameters, a straightforward and easy-to-implement structure, and high computational efficiency. As a result, MPA has been widely applied in academic optimization and industrial applications. However, MPA still suffers from shortcomings, such as precocious convergence, poor searchability, and local optimality. To improve the performance of MPA, many researchers have devoted themselves to enhancing its adaptability and scalability. Salih [14] introduced an enhanced MPA named Dynamic Opposition and Taylor Neighborhood Search MPA (DOTMPA). Key improvements in DOTMPA include the Taylor-based Optimal Neighborhood Strategy (TNS), which mitigates the risk of local optima entrapment and premature convergence, and an Asymmetric Search Space with Dynamic Option-based Learning integrated into DOTMPA increases the probability of identifying the global optimum. DOTMPA was tested against other optimization algorithms on CEC2017, CEC2019, and CEC2022 benchmarks and demonstrated superior performance. It was also applied to engineering problems such as cantilever beams, springs, speed reducers, and structural optimization

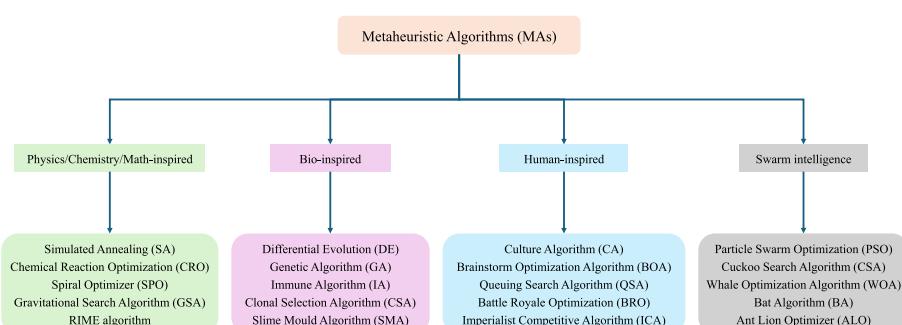


Fig. 1 Classification of MAs

tasks, proving to be highly accurate, stable, and competitive in real-world applications. Kumar et al. [15] introduced the Chaotic MPA (CMPA) to optimize engineering problems by combining MPA with chaotic maps to enhance search control. CMPA was tested on the CEC 2020 benchmark, constrained design tasks, multi-pass turning case studies, and structural topology optimization, demonstrating significant improvements compared to twelve competitor algorithms, including MPA variants and CEC competition winners, proving its practical effectiveness in tackling complex design challenges. Chen et al. [16] proposed a hybrid MPA variant with Q-learning (QMPA), where reinforcement learning dynamically selects the optimal update strategy for search agents. QMPA improved the adaptability and optimization performance of MPA and outperformed MPA on classical benchmark functions, CEC2014, and engineering problems. It also surpassed six other algorithms in the CEC2014 test suite and achieved optimal results in two real-world engineering problems enhancing both convergence and stability. Zhang et al. [17] developed the Optimized Dynamic MPA (ODMPA), which addresses the limitations of MPA using a tent map, outpost mechanism, and DE-SA mutation. These elements improve exploration by enhancing agent diversity and boosting convergence speed. ODMPA excelled in CEC2014 benchmarks, engineering problems, and photovoltaic tasks and outperformed other algorithms in terms of accuracy and effectiveness in solving optimization problems, which demonstrates its competitiveness and effectiveness. Han et al. [18] introduced the Golden-Sine Dynamic MPA (GDMPA), which incorporates Logistic-Logistic cascade chaos for high-quality initial population, a dynamic transition probability for balancing exploration and exploitation, and adaptive inertial weight to prevent local optimality. The Golden-Sine factor further improved convergence and population diversity. Comparison experiments on CEC2008, CEC2017, and truss structure optimization confirmed that GDMPA outperformed MPA and other algorithms, which significantly reduces the computational budget in truss optimization and highlights its potential in structural design. Hu et al. [19] developed the Neighborhood-based MPA (NMPA) with a neighborhood learning strategy and adaptive population size, which led to better performance on CEC benchmarks and engineering design problems. NMPA demonstrated improved precision, convergence, and solution quality and successfully addressed real-world problems, such as developable surface modeling, which underscores its potential in engineering optimization. Oszust et al. [20] introduced the LEO-MPA, an MPA variant with a Local Escaping Operator (LEO) that replaces the worst candidates with better solutions, thus enhancing population diversity. LEO-MPA was tested on 82 functions, including CEC2014, CEC2017, and various engineering problems, consistently outperforming MPA and other algorithms. The study also explored the hybridization of MAs with LEO and demonstrated significant improvements in optimization performance. Zhong et al. [21] introduced a hybrid MPA with Teaching-Learning-Based Optimization TLBO to form TLMPA, which combines MPA with the TLBO framework. TLMPA incorporates effective mutation and crossover strategies to improve predator diversity and prevent premature convergence. Numerical experiments on CEC2017 and engineering design problems confirmed that TLMPA achieved superior performance compared to state-of-the-art MAs. Houssein et al. [22] developed the Opposition-Based Learning-enhanced MPA (MPA-OBL), which improves search efficiency and convergence speed. Extensive experiments on CEC2020

benchmark problems demonstrated the effectiveness of MPA-OBL, which outperformed various algorithms such as LSHADE-SPACMA-OBL and CMA-ES-OBL. It was further validated through image segmentation challenges using Otsu and Kapur's methods, and evaluated with PSNR, SSIM, and FSIM indices, confirming the reliability and efficiency of MPA-OBL.

Despite the widespread applicability of MPA across various optimization domains, it still faces several critical issues that limit its performance and robustness, such as insufficient uniformity in population initialization, unbalanced exploration and exploitation, and premature convergence due to inadequate population diversity. Focusing on these issues, this paper proposes a Multi-strategy Enhanced MPA (MEMPA), where effective strategies are integrated to improve the comprehensive performance of MPA in various optimization challenges. Specifically, three efficient search operators are integrated into the proposed MEMPA (1) Sobol sequence-based swarm initialization to construct a more uniformly distributed population in the initialization phase, (2) mutualism mechanism in the exploitation phase to enhance the exploitative searchability, and (3) distance-based selection scheme to maintain the diversity of the population. The proposed MEMPA is comprehensively investigated in numerical experiments including CEC2020, engineering problems, and intrusion detection challenges. Specifically, the main contributions of this paper are as follows:

- In the initialization stage, the Sobol sequence is used instead of uniform random to generate more uniform solutions in the high-dimensional search space and improve the diversity of the initial population.
- The mutualism mechanism is introduced to increase the exploitation ability of MEMPA.
- The distance selection mechanism is introduced to allow the suboptimal solution to survive probabilistically, and the population diversity can be well maintained to strengthen the ability to escape local optimality.
- The proposed MEMPA is evaluated on CEC2020 and six engineering problems against eleven well-known MAs.
- We integrate MEMPA with the Extreme Learning Machine (ELM) and apply MEMPA-ELM to intrusion detection.

The remainder of this paper is structured as follows: Sect. [Marine predator algorithm \(MPA\)](#) introduces MPA. Section [Multi-strategy enhanced marine predator algorithm \(MEMPA\)](#) details the proposed MEMPA. Numerical experiments on CEC2020 are conducted in Sect. [Numerical experiments](#), and the engineering optimization and MEMPA-ELM for intrusion detection are presented in Sects. [Experimental results on engineering problems](#) and [Experiments on intrusion detection](#), respectively. Finally, section [Conclusion and future works](#) concludes this paper.

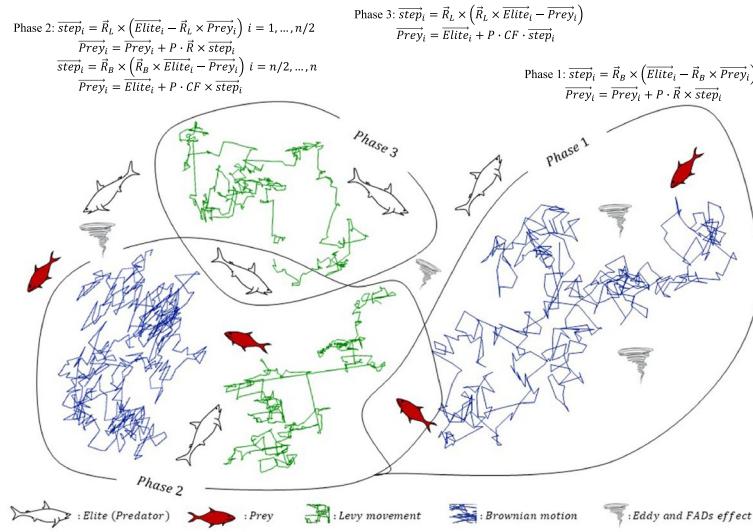


Fig. 2 A demonstration of MPA [13]

Marine predator algorithm (MPA)

MPA simulates predatory behaviors of large marine predators such as sharks and whales in marine ecosystems, and a demonstration of MPA is presented in Fig. 2.

Here, the top predator, denoted as *Elite*, is the most powerful. Each position in *Elite* provides information about the location of prey found so far for the subsequent foraging behavior. Equation (1) define the matrix *Elite*.

$$Elite = \begin{bmatrix} X_{11}^I & X_{12}^I & \cdots & X_{1D}^I \\ X_{21}^I & X_{22}^I & \cdots & X_{2D}^I \\ X_{31}^I & X_{32}^I & \cdots & X_{3D}^I \\ \vdots & \vdots & \ddots & \vdots \\ X_{N1}^I & X_{N2}^I & \cdots & X_{ND}^I \end{bmatrix} \quad (1)$$

where N and D denote the number of predators and the dimension size, respectively. X^I represents the top-level predator, where *Elite* is constructed by copying X^I with N times. In the end, if a predator with better fitness is found, it will be the new top predator and the *Elite* matrix will be updated. *Prey* is another matrix that has an identical scale to *Elite*, which is utilized to update the position of predators as defined in Eq. (2).

$$Prey = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1D} \\ X_{21} & X_{22} & \cdots & X_{2D} \\ X_{31} & X_{32} & \cdots & X_{3D} \\ \vdots & \vdots & \ddots & \vdots \\ X_{N1} & X_{N2} & \cdots & X_{ND} \end{bmatrix} \quad (2)$$

where X_{ij} denotes the value of i^{th} predator in the j^{th} dimension. The optimization procedures of MPA are further divided into phases: the high-speed phase, the unit-speed phase, and the low-speed phase.

Phase 1: In the high-speed phase, the Brownian movement is employed to describe the behavior of predators in marine ecosystems as formulated in Eq. (3).

$$\begin{aligned} \vec{\text{step}}_i &= \vec{R}_B \times (\vec{\text{Elite}}_i - \vec{R}_B \times \vec{\text{Prey}}_i) \\ \vec{\text{Prey}}_i &= \vec{\text{Prey}}_i + P \cdot \vec{R} \times \vec{\text{step}}_i \end{aligned} \quad (3)$$

where $\vec{\text{Elite}}_i$ and $\vec{\text{Prey}}_i$ indicate the i^{th} individual in the matrix Elite and Prey , respectively. \vec{R}_B is a D -dimensional vector where each element is generated by a standard normal distribution to represent the Brownian movement. $P=0.5$ is a scaling constant, \vec{R} is a random vector where each element is in $[0, 1]$, and \times denotes the entry-wise multiplication.

Phase 2: In the unit-speed phase, the prey and predator have the same speed, and the position of predators is updated using Eq. (4).

$$\begin{aligned} \vec{\text{step}}_i &= \vec{R}_L \times (\vec{\text{Elite}}_i - \vec{R}_L \times \vec{\text{Prey}}_i), \quad i = 1, \dots, N/2 \\ \vec{\text{Prey}}_i &= \vec{\text{Prey}}_i + P \cdot \vec{R} \times \vec{\text{step}}_i \\ \vec{\text{step}}_i &= \vec{R}_B \times (\vec{R}_B \times \vec{\text{Elite}}_i - \vec{\text{Prey}}_i), \quad i = N/2, \dots, N \\ \vec{\text{Prey}}_i &= \vec{\text{Elite}}_i + P \cdot CF \cdot \vec{\text{step}}_i \end{aligned} \quad (4)$$

where \vec{R}_L is a D -dimensional random vector, and each element is generated by the Lévy flight generator. CF is a control parameter to affect the movement scale, as defined in Eq. (5).

$$CF = (1 - \frac{t}{T})^{2\frac{t}{T}} \quad (5)$$

where t is the current iteration and T is the maximum iteration.

Phase 3: In the low-speed phase, Eq. (6) is employed to describe the movement of predators.

$$\begin{aligned} \vec{\text{step}}_i &= \vec{R}_L \times (\vec{R}_L \times \vec{\text{Elite}}_i - \vec{\text{Prey}}_i) \\ \vec{\text{Prey}}_i &= \vec{\text{Elite}}_i + P \cdot CF \cdot \vec{\text{step}}_i \end{aligned} \quad (6)$$

Additionally, the formation of eddies (*FADs*) in nature can influence the foraging behaviors of marine predators. MPA also considers this factor and formulates FADs using Eq. (7).

$$\vec{\text{Prey}}_i = \begin{cases} \vec{\text{Prey}}_i + CF(\vec{X}_{min} + \vec{R}_i \times (\vec{X}_{max} - \vec{X}_{min})) \times \vec{U}, & \text{if } r \leq FADs \\ \vec{\text{Prey}}_i + (FADs \cdot (1 - r) + r) \cdot (\vec{\text{Prey}}_{r1} - \vec{\text{Prey}}_{r2}), & \text{otherwise} \end{cases} \quad (7)$$

where $FADs=0.2$ represents the probability of *FADs* effect happening, r is a random number within $(0, 1)$, V is a binary vector within 0 or 1, \vec{X}_{max} and \vec{X}_{min} denote the upper and lower bounds of search domain, respectively. In summary, the pseudocode of MPA is presented in Algorithm 1.

Algorithm 1 MPA

Require: Dimension: D , Population size: PS , Maximum iteration: T_{max} ,
Ensure: Global optimum: GO

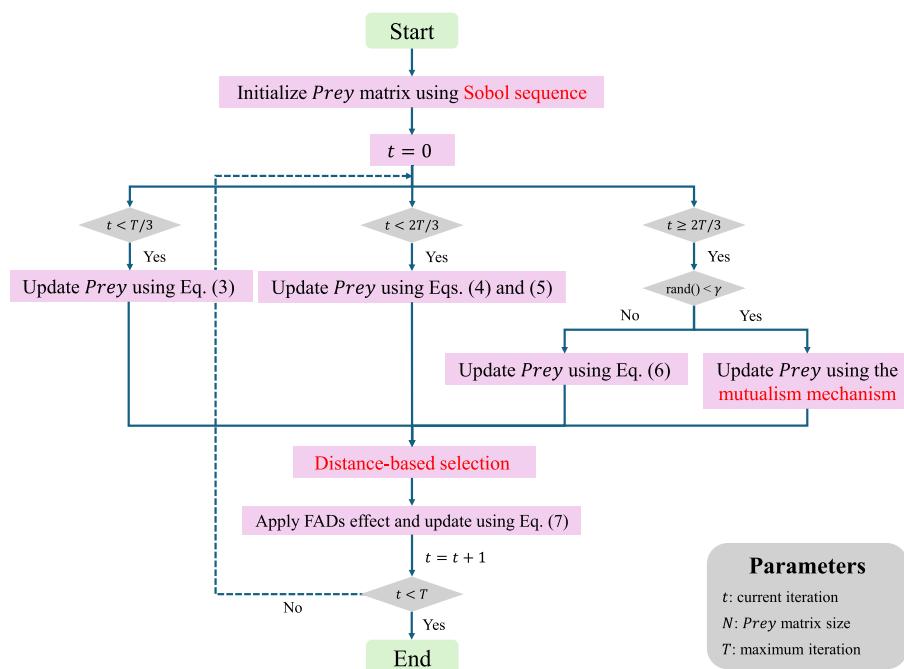
```

1: Initialize the predator matrix  $Prey$ .
2: Record the current best individual  $GO$ .
3:  $t = 0$ 
4: while  $t < T_{max}$  do
5:   Initialize top-predator matrix  $Elite$ .
6:   if  $t < T/3$  then
7:     Update  $Prey$  using Eq. (3)
8:   else if  $t < 2T/3$  then
9:     Update  $Prey$  using Eqs. (4) and (5)
10:  else
11:    Update  $Prey$  using Eq. (6)
12:  end if
13:  Apply FADs effect using Eq. (7)
14:  Record the current best individual  $GO$ .
15:   $t = t + 1$ 
16: end while
17: return  $GO$ 

```

Multi-strategy enhanced marine predator algorithm (MEMPA)

This section starts to introduce our proposed Multi-strategy Enhanced MPA (MEMPA) by presenting the flowchart in Fig. 3. The proposed MEMPA inherits the main framework of MPA and integrates three efficient strategies: the Sobol sequence initialization, the mutualism mechanism, and the distance-based selection. The details of these strategies are introduced in the following contexts.

**Fig. 3** The flowchart of MEMPA

Sobol sequence initialization

The allocation of initialization populations may significantly influence the convergence of MAs. When dealing with unknown problems, the initial population is required to be uniformly distributed in the search domain to ensure population diversity. However, the random number generator in traditional MAs may cause the aggregation of samples in the search space, which significantly affects the optimization performance of MAs [23].

Low-discrepancy sequences aim to fill a multidimensional unit hypercube with points as uniformly as possible by selecting an appropriate sampling direction. This approach offers higher efficiency and uniformity when addressing probability problems. Among the various low-discrepancy sequences, the Sobol sequence is particularly advantageous due to its shorter computation cycle and faster generation of points [24]. Therefore, the motivation of this research is to utilize the Sobol sequence to generate a high-quality initial population and accelerate the optimization while maintaining the population diversity.

Assuming that the boundary of the search domain is \vec{X}_{min} and \vec{X}_{max} , the range of random value k_n generated by the Sobol sequence is within $[0, 1]$, and the value of i^{th} predator in the j^{th} dimension can be calculated using Eq. (8).

$$X_{ij} = \vec{X}_{min} + k_n \cdot (\vec{X}_{max} - \vec{X}_{min}) \quad (8)$$

To demonstrate the ability of the Sobol sequence to generate more uniformly distributed random sequences, we randomly generated 10,000 values using both the Sobol sequence and the uniform distribution. Figure 4 demonstrates the distribution of these generated points, where the horizontal axis represents the numerical values and the vertical axis indicates the frequency with which a value falls within a specific range. Compared to sequences generated by a uniform distribution, the random numbers generated by the Sobol sequence tend to be more uniformly distributed in high-dimensional search spaces.

Mutualism mechanism

Cheng and Prayogo [25] proposed the Symbiotic Organisms Search (SOS) algorithm, which simulates the interactions between organisms within an ecosystem. In SOS, the population represents an ecosystem where individuals are updated through three

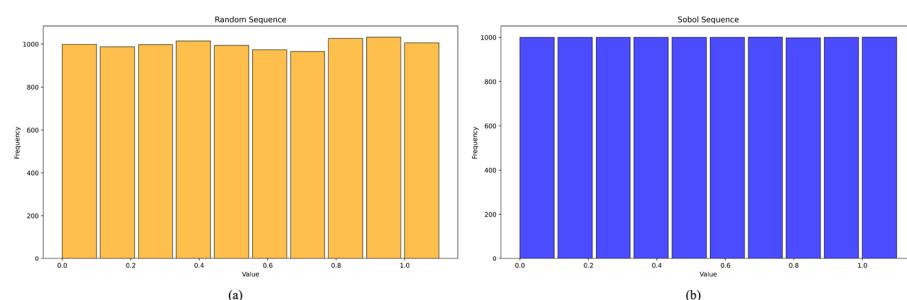


Fig. 4 Bar figures of the Sobol sequence and the uniform distribution

types of interactions: mutualism, commensalism, and parasitism. This reciprocal process allows SOS to have strong exploitation capabilities. Equation (9) demonstrates the update principle for the mutualism mechanism in SOS.

$$\begin{aligned} \vec{Mutual} &= \frac{\vec{X}_{i1}^t + \vec{X}_{i2}^t}{2} \\ \vec{X}_{i1}^{t+1} &= \vec{X}_{i1}^t + r \cdot (\vec{X}_{best}^t - \vec{Mutual} \cdot BF_1) \\ \vec{X}_{i2}^{t+1} &= \vec{X}_{i2}^t + r \cdot (\vec{X}_{best}^t - \vec{Mutual} \cdot BF_2) \end{aligned} \quad (9)$$

where $Mutual$ denotes the relationship between the two organisms \vec{X}_{i1}^t and \vec{X}_{i2}^t , r is a random value within $[0, 1]$, \vec{X}_{best}^t represents the optimum found so far, BF_1 and BF_2 are the interest factors which randomly selected from $\{1, 2\}$. To accelerate the optimization and improve the scalability of MAP, we integrate the mutualism mechanism into the **Phase 3**, as formulated in Eq. (10).

$$\begin{aligned} \vec{Mutual} &= \frac{\vec{Prey}_{i1}^t + \vec{Prey}_{i2}^t}{2} \\ \vec{Prey}_{i1}^{t+1} &= \vec{Prey}_{i1}^t + P \cdot CF \cdot \vec{R}_L \times (\vec{R}_L \times \vec{Prey}_{best}^t - \vec{Mutual} \cdot BF_1) \\ \vec{Prey}_{i2}^{t+1} &= \vec{Prey}_{i2}^t + P \cdot CF \cdot \vec{R}_L \times (\vec{R}_L \times \vec{Prey}_{best}^t - \vec{Mutual} \cdot BF_2) \end{aligned} \quad (10)$$

where \vec{R}_L is a random vector generated from the Lévy flight distribution and \vec{Prey}_{best}^t is the current best predator. The mutualism mechanism from SOS can be seamlessly integrated into the low-speed phase of MPA, which allows MEMPA to benefit from this hybridization by reducing the probability of stagnation in the later stages of optimization.

Distance-based selection

Similar to the Simulated Annealing (SA), the distance-based selection proposed in [26] allows inferior offspring individuals to have a probability of being accepted and survive to the next iteration. The fundamental difference between these two schemes is that the acceptance probability for inferior individuals in SA will reduce as the optimization promotes, while distance-based selection determines the acceptance probability based on the distance between parent and offspring individuals [27]. Equation (11) presents the principle of the distance-based selection for the minimization problems.

$$\vec{X}_i^{t+1} = \begin{cases} \vec{O}_i^t, & \text{if } f(\vec{O}_i^t) \leq f(\vec{X}_i^t) \\ \vec{O}_i^t, & \text{else if } r < e^{-\frac{\Delta f}{Dis}} \\ \vec{X}_i^t, & \text{otherwise} \end{cases} \quad (11)$$

where \vec{X}_i^t and \vec{O}_i^t denote the i^{th} parent individuals in t^{th} iteration and its offspring individual, respectively. r is a random number within $[0, 1]$, e is the base of the natural logarithm, $\Delta f = |f(\vec{O}_i^t) - f(\vec{X}_i^t)|$ is the absolute fitness difference between \vec{X}_i^t and \vec{O}_i^t , and Dis denotes the Manhattan distance between \vec{X}_i^t and \vec{O}_i^t as formulated in Eq. (12).

$$Dis = \sum_{j=1}^D |\vec{X}_{ij}^t - \vec{O}_{ij}^t| \quad (12)$$

Generally, three criteria consist of the distance-based selection mechanism.

- **Case 1:** When the offspring individual \vec{O}_i^t has a better fitness value than its parent individual \vec{X}_i^t , it will directly survive to the next iteration.
- **Case 2:** When the offspring individual \vec{O}_i^t has a worse fitness value than its parent individual \vec{X}_i^t , the distance-based selection mechanism generates a random value r and compares it with $e^{-\frac{\Delta f}{Dis}}$. If this condition is satisfied, \vec{O}_i^t is still accepted and survives.
- **Case 3:** If the previous two conditions are unsatisfied, \vec{O}_i^t will be abandoned and \vec{X}_i^t will survive.

Based on the formulation in Eq. (12), when the fitness difference between \vec{O}_i^t and \vec{X}_i^t is smaller or the distance between \vec{O}_i^t and \vec{X}_i^t is greater, the value of $e^{-\frac{\Delta f}{Dis}}$ increases, and the probability of \vec{O}_i^t being accepted is higher. Consequently, the integration of the distance-based selection mechanism encourages MEMPA to explore the unknown search space and prevent MEMPA from falling into local optima. In summary, the pseudocode of MEMPA is presented in Algorithm 2.

Algorithm 2 MEMPA

Require: Dimension: D , Population size: PS , Maximum iteration: T_{max} ,
Ensure: Global optimum: GO

```

1: Initialize the predator matrix  $Prey$  using the Sobol sequence.
2: Record the current best individual  $GO$ .
3:  $t = 0$ 
4: while  $t < T_{max}$  do
5:   Initialize top-predator matrix  $Elite$ .
6:   if  $t < T/3$  then
7:     Update  $Prey$  using Eq. (3)
8:   else if  $t < 2T/3$  then
9:     Update  $Prey$  using Eqs. (4) and (5)
10:   else
11:     if  $rand() < \gamma$  then
12:       Update  $Prey$  using Eq. (10)
13:     else
14:       Update  $Prey$  using Eq. (6)
15:     end if
16:   end if
17:   Apply FADs effect using Eq. (7)
18:   Distance-based selection using Eq. (12)
19:   Record the current best individual  $GO$ .
20:    $t = t + 1$ 
21: end while
22: return  $GO$ 

```

Computational complexity of MEMPA

Supposing that the population size of marine predators is N , the dimension of the search space D , and the maximum number of iterations is T . The computational complexity of the original MPA is $O(N \cdot T \cdot D)$ as analyzed in [13], and the computational complexity of additional strategies is analyzed as follows.

- Marine predators initialization using the Sobol sequence: $O(N \cdot D)$.
- Symbiosis mechanism: $O(2N \cdot D)$.
- Distance-based selection mechanism: $O(N \cdot D)$.

Therefore, the computational complexity of MEMPA is $O(N \cdot T \cdot D + T \cdot (2N \cdot D + N \cdot D)) := O(N \cdot T \cdot D)$. Although the theoretical computational analysis between MPA and MEMPA is the same, the practical computational complexity has an acceptable difference.

Numerical experiments

Experimental environments and implementation

To comprehensively investigate the performance of the proposed MEMPA, we conduct numerical experiments in CEC2020, engineering problems, and intrusion detection challenges. Eleven well-known MAs are employed as competitor algorithms to evaluate the performance of the proposed MEMPA: Coyote Optimization Algorithm (COA) [28], Marine predator algorithm (MPA) [13], Artificial Gorilla Troops Optimization (AGTO) [29], Arithmetic Optimization Algorithm (AOA) [30], Serval Optimization Algorithm (SOA) [31], Zebra Optimization Algorithm (ZOA) [32], Golden jackal optimization (GJO) [33], Energy Valley Optimizer (EVO) [34], Improved MPA (IMPA) [35], Enhanced MPA (EMPA) [36], and Chaotic MPA (CMPA) [15]. The parameter settings of these algorithms are summarized in Table 1.

The parameter settings of optimizers are consistent with the optimal configuration suggested in the corresponding publications. Additionally, the population size and maximum number of iterations are fixed at 30 and 1000 as recommended in [37–39]. Each algorithm is repeated 30 times to alleviate the effect of randomness.

Table 1 The parameter setting of algorithms

MAs	Parameters	Value
COA	Hyperparameter-free	
MPA	Scaling factor	0.5
	Crossover probability	0.2
AGTO	Transition probability in exploration	0.03
	Transition probability in exploitation	0.8
	Coefficient	3.0
AOA	Sensitivity parameter	5
	Min. and Max. accelerator	0.2 and 0.9
SOA	Hyperparameter-free	
ZOA	Hyperparameter-free	
GJO	Hyperparameter-free	
EVO	Hyperparameter-free	
IMPA	Scaling factor	0.5
	Crossover probability	0.2
EMPA	Scaling factor	0.5
CMPA	Chaos type	Logistic map
MEMPA	Scaling factor	0.5
	Crossover probability	0.2

The competitor algorithms used for comparison in this study are sourced from the MEALPY library [40], which provides a comprehensive suite of state-of-the-art MAs.

Comparison experiments on CEC2020

This section conducts the comparison experiments on CEC2020. Tables 2, 3, 4, and 5 summarize the experimental results and statistical analysis on CEC2020 with various dimensions, and the best value is in bold, while convergence curves are presented in Figs. 5, 6, 7, and 8.

The functions in the CEC2020 suite exhibit a wide range of characteristics, including shifts, rotations, and a mix of unimodal, multimodal, and composite structures. This diversity makes the suite well-suited for comprehensively evaluating the overall performance of our proposed MEMPA algorithm. In the unimodal function F1, MEMPA demonstrates highly competitive performance among the eleven optimizers, which showcases a strong exploitation ability and outperforms all competitor algorithms across all dimensions. Notably, as the dimension of the problem increases, the difference in fitness values between MEMPA and other optimizers becomes more obvious, further emphasizing the superiority of MEMPA in handling high-dimensional problems.

Moreover, the presented optimization curves confirm that during the slower exploitation stages, the integration of the mutualism mechanism allows the optimization curve to decline more rapidly in most functions, which underscores the enhancement in the exploitation ability of MEMPA. This demonstrates that the introduction of the mutualism mechanism can accelerate optimization convergence and further improve the performance of MEMPA.

The rest of the functions F2 to F9 are multimodal, hybrid, and composite functions, characterized by complex fitness landscapes. These functions are valuable for investigating the ability of optimizers to escape local optima and explore the search space efficiently. Based on the summarized results, where the average rank of MEMPA improves from 3.5 to 1.8, 1.4, and 1.0 as the dimensionality of the problems increases. This trend further indicates the advantage of MEMPA in tackling high-dimensional problems and superior performance in most cases.

Ablation experiments on CEC2020

This section aims to investigate the contribution of different combinations of the three integrated strategies. Here, we define the abbreviations in the experiment as: MPA-ss: MPA + Sobol sequence initialization; MPA-mm: MPA + symbiosis mechanism; MPA-dsm: MPA + distance-based selection mechanism; MPA-ss+mm: MPA + Sobol sequence initialization + symbiosis mechanism; MPA-ss+dsm: MPA + Sobol sequence initialization + distance-based selection mechanism; MPA-mm+dsm: MPA + symbiosis mechanism + distance-based selection mechanism. The original MPA serves as the primary approach for statistical analysis. Experimental results on CEC2020 are summarized in Tables 6, 7, 8, and 9.

The experimental results and statistical analysis show that the population initialization using the Sobol sequence has the effect of accelerating the optimization convergence in the case of low dimensions. For example, in the 10-D problem, MPA-ss achieved

Table 2 Experimental results and statistical analysis on 10-D CEC2020

Func.	COA	MPA	AGTO	AOA	SOA	ZOA	GIO	EVO	IMPA		CMPA	MEMPA
									EMPA	IMPA		
F1	Mean	1.145e+08 + 1.955e+03 \approx	3.742e+03 +	1.647e+10 +	9.909e+09 +	1.483e+09 +	9.226e+08 +	2.687e+09 +	3.687e+03 +	2.977e+03 +	1.912e+03 \approx	1.523e+03
	Std	4.779e+07	1.941e+03	3.100e+03	5.946e+09	2.229e+09	9.822e+08	2.069e+08	2.201e+09	3.268e+03	2.793e+03	2.468e+03
F2	Mean	1.664e+10 + 2.073e+05 \approx	3.275e+05 +	1.249e+12 +	6.110e+11 +	7.779e+10 +	1.165e+11 +	1.901e+11 +	2.163e+05 \approx	1.745e+05 \approx	3.754e+05 +	1.777e+05
	Std	8.026e+09	2.519e+05	4.390e+05	4.387e+11	2.315e+11	4.473e+10	3.195e+10	1.498e+11	3.326e+05	2.258e+05	3.184e+05
F3	Mean	2.967e+09 + 8.204e+04 \approx	1.314e+05 \approx	3.906e+11 +	1.990e+11 +	2.199e+10 +	5.132e+10 +	5.796e+10 +	1.251e+05 \approx	1.128e+05 \approx	1.114e+05 \approx	1.215e+05
	Std	1.439e+09	1.008e+05	1.387e+05	5.578e+10	5.868e+10	1.247e+10	2.529e+10	4.778e+10	1.339e+05	1.290e+05	1.346e+05
F4	Mean	1.906e+03 + 1.901e+03 \approx	1.904e+03 +	1.030e+05 +	3.980e+03 +	2.012e+03 +	1.943e+03 +	1.346e+04 +	1.901e+03 -	1.901e+03 -	1.901e+03 -	1.901e+03
	Std	8.455e-01	4.002e-01	2.000e+00	1.139e+05	2.218e+03	1.875e+02	6.786e+01	5.910e+04	3.912e+01	3.867e+01	3.287e+01
F5	Mean	2.996e+04 + 2.259e+03 +	3.604e+03 +	3.428e+07 +	1.199e+05 +	1.328e+05 +	2.672e+05 +	8.054e+05 +	2.106e+03 \approx	2.547e+03 +	2.110e+03 \approx	2.023e+03
	Std	9.169e+03	3.625e+02	5.080e+03	3.284e+07	6.638e+04	1.279e+05	1.739e+05	1.858e+06	3.091e+02	5.645e+02	3.581e+02
F6	Mean	3.529e+03 + 1.610e+03 \approx	1.773e+03 +	9.684e+07 +	3.337e+03 +	5.911e+03 +	4.152e+03 +	4.271e+05 +	1.608e+03 -	1.613e+03 -	1.615e+03 -	1.621e+03
	Std	6.984e+02	1.831e+01	1.796e+02	1.544e+08	9.763e+02	4.289e+03	4.352e+01	1.142e+06	1.340e+01	1.561e+01	2.482e+01
F7	Mean	5.562e+04 + 2.528e+03 +	5.043e+03 +	5.678e+07 +	2.246e+04 +	6.079e+04 +	9.114e+04 +	1.734e+06 +	2.470e+03 +	2.787e+03 +	2.476e+03 +	2.389e+03
	Std	2.402e+04	2.750e+02	6.399e+03	7.717e+07	5.126e+03	1.228e+05	6.281e+04	6.832e+06	2.450e+02	4.759e+02	2.768e+02
F8	Mean	2.312e+03 + 2.301e+03 -	2.306e+03 +	2.517e+03 +	2.341e+03 +	2.320e+03 +	2.316e+03 +	2.325e+03 +	2.301e+03 -	2.302e+03 -	2.302e+03 -	2.305e+03
	Std	9.904e-01	1.133e+00	2.143e+00	1.858e+02	1.793e+01	5.901e+00	5.360e+00	9.310e+00	1.113e+00	1.144e+00	1.298e+00
F9	Mean	3.061e+03 + 2.572e+03 \approx	2.582e+03 -	9.972e+03 +	5.816e+03 +	3.927e+03 +	3.785e+03 +	4.342e+03 +	2.560e+03 -	2.576e+03 -	2.552e+03 \approx	2.610e+03
	Std	2.086e+02	5.674e+01	6.682e+01	1.659e+03	1.063e+03	5.144e+02	6.986e+02	8.808e+02	4.897e+01	5.159e+01	5.856e+01
F10	Mean	3.005e+03 + 2.979e+03 -	3.006e+03 +	3.469e+03 +	3.262e+03 +	3.098e+03 +	3.056e+03 +	3.165e+03 +	2.979e+03 -	2.981e+03 -	2.980e+03 -	2.989e+03
	Std	1.391e+01	1.526e+00	3.687e+01	1.916e+02	7.217e+01	5.724e+01	1.939e+01	8.084e+01	1.562e+00	8.777e+00	6.050e+00
Avg/ranks:		10/0/0	2/6/2	8/1/1	10/0/0	10/0/0	10/0/0	10/0/0	2/3/5	3/2/5	2/4/4	-
Avg/ranks:	7.1	2.4	5.8	12.0	9.8	8.9	8.7	10.4	2.5	3.5	3.4	35

Table 3 Experimental results and statistical analysis on 30-D CEC2020

Func.	COA	MPA	AGTO	AOA	SOA	ZOA	GJO	EVO	IMPA	EMPA	CMPA	MEMPA
F1	Mean	1.101e+10 + 4.511e+04 +	4.440e+05 +	6.222e+10 +	5.408e+10 +	2.233e+10 +	2.236e+10 +	2.391e+10 +	5.377e+04 +	2.134e+05 +	5.398e+04 +	2.390e+04
	Std	4.308e+09 2.907e+04	1.009e+06 5.476e+09	8.162e+09 4.357e+09	3.354e+09 7.781e+09	3.612e+04 3.354e+09	3.612e+04 7.781e+09	1.503e+05 3.428e+12 +	2.432e+04 2.086e+06 +	1.974e+04 2.086e+07 +	2.432e+04 3.108e+06 +	1.870e+06
F2	Mean	1.106e+12 + 4.218e+06 +	7.137e+08 +	8.079e+12 +	6.223e+12 +	3.121e+12 +	2.288e+12 +	2.786e+12 +	2.786e+12 +	2.086e+07 +	3.108e+06 +	2.086e+06
	Std	4.840e+11 3.277e+06	3.551e+09 8.630e+11	1.009e+12 6.097e+11	2.984e+11 7.666e+11	6.097e+11 7.666e+11	2.984e+11 7.666e+11	2.902e+06 2.113e+07	2.234e+06 2.113e+07	2.234e+06 2.113e+07	2.234e+06 2.454e+06	2.454e+06
F3	Mean	3.763e+11 + 3.310e+06 +	6.482e+07 +	2.398e+12 +	1.89e+12 +	8.285e+11 +	7.891e+11 +	8.629e+11 +	3.362e+06 +	1.442e+07 +	2.823e+06 +	1.181e+06
	Std	1.146e+11 2.219e+06	2.776e+08 3.038e+11	3.684e+11 1.870e+11	1.870e+11 1.145e+11	2.657e+11 2.405e+06	2.657e+11 2.405e+06	2.405e+06 1.259e+07	2.384e+06 2.384e+07	2.384e+06 6.255e+05	2.384e+06 6.255e+05	6.255e+05
F4	Mean	5.494e+03 + 1.908e+03 ≈	2.002e+03 +	2.629e+06 +	7.224e+05 +	1.381e+05 +	3.092e+04 +	1.033e+05 +	1.908e+03 +	1.908e+03 +	1.908e+03 +	1.908e+03 –
	Std	5.392e+03 2.254e+00	7.077e+01 8.199e+05	2.994e+05 1.022e+05	1.312e+04 9.869e+04	1.312e+04 9.869e+04	1.312e+04 9.869e+04	2.007e+00 3.788e+00	2.407e+00 3.788e+00	2.407e+00 3.788e+00	2.319e+00 2.319e+00	2.319e+00
F5	Mean	1.150e+07 + 7.668e+04 ≈	1.598e+05 + 4.822e+08 +	1.240e+08 +	3.854e+07 +	8.476e+06 +	5.553e+07 +	8.542e+04 +	1.079e+05 +	8.212e+04 +	8.212e+04 +	7.555e+04
	Std	4.987e+06 1.978e+04	1.216e+05 1.892e+08	5.163e+07 3.772e+07	3.772e+07 3.770e+06	5.876e+07 2.739e+04	5.876e+07 2.739e+04	5.358e+04 5.358e+04	3.341e+04 3.341e+04	3.341e+04 3.128e+04	3.341e+04 3.128e+04	3.128e+04
F6	Mean	4.131e+05 + 2.979e+03 ≈	1.175e+04 + 1.469e+09 +	2.186e+05 + 4.281e+06 +	5.754e+06 + 5.754e+06 +	3.931e+07 + 3.931e+07 +	2.501e+03 ≈ 5.394e+03 +	5.394e+03 + 5.394e+03 +	2.605e+03 + 2.605e+03 +	2.510e+03 + 2.510e+03 +	2.510e+03 + 2.510e+03 +	2.510e+03
	Std	3.366e+05 1.436e+03	1.511e+04 7.071e+08	1.219e+05 9.377e+05	1.574e+06 1.574e+06	7.025e+07 7.025e+07	8.054e+02 8.054e+02	3.733e+03 3.733e+03	9.471e+02 9.471e+02	7.940e+02 7.940e+02	7.940e+02 7.940e+02	7.940e+02
F7	Mean	2.867e+07 + 1.476e+05 +	1.768e+05 + 4.699e+09 +	5.879e+08 + 7.098e+07 +	5.195e+07 + 5.195e+07 +	1.598e+08 + 1.598e+08 +	1.232e+05 + 1.232e+05 +	2.470e+05 + 2.470e+05 +	1.551e+05 + 1.551e+05 +	1.019e+05 + 1.019e+05 +	1.019e+05 + 1.019e+05 +	1.019e+05
	Std	1.739e+07 9.516e+04	1.735e+05 4.332e+08	5.196e+07 5.196e+07	1.316e+07 1.316e+07	1.491e+08 1.491e+08	5.704e+04 5.704e+04	2.005e+05 2.005e+05	8.865e+04 8.865e+04	5.613e+04 5.613e+04	5.613e+04 5.613e+04	5.613e+04
F8	Mean	2.527e+03 + 2.379e+03 –	2.459e+03 + 6.593e+03 +	3.559e+03 + 3.030e+03 +	2.661e+03 + 2.661e+03 +	2.686e+03 + 2.686e+03 +	2.380e+03 – 2.380e+03 –	2.383e+03 – 2.383e+03 –	2.381e+03 – 2.381e+03 –	2.397e+03 – 2.397e+03 –	2.397e+03 – 2.397e+03 –	2.397e+03
	Std	2.728e+01 5.733e+00	4.977e+01 9.816e+02	1.950e+02 2.269e+02	8.542e+01 1.592e+02	7.468e+00 7.468e+00	8.652e+00 8.652e+00	5.208e+00 5.208e+00	1.319e+01 1.319e+01	1.319e+01 1.319e+01	1.319e+01 1.319e+01	1.319e+01
F9	Mean	1.105e+04 + 2.613e+03 +	3.114e+03 + 3.660e+04 +	3.250e+04 + 2.287e+04 +	1.789e+04 + 1.789e+04 +	2.218e+04 + 2.218e+04 +	2.614e+03 + 2.614e+03 +	2.639e+03 + 2.639e+03 +	2.612e+03 + 2.612e+03 +	2.607e+03 + 2.607e+03 +	2.607e+03 + 2.607e+03 +	2.607e+03
	Std	2.318e+03 3.178e+00	1.195e+03 1.434e+03	1.359e+03 2.207e+03	5.871e+03 5.871e+03	5.188e+00 5.188e+00	5.768e+01 5.768e+01	3.979e+00 3.979e+00	3.979e+00 3.979e+00	3.979e+00 3.979e+00	3.979e+00 3.979e+00	3.979e+00
F10	Mean	3.393e+03 + 2.928e+03 ≈	2.956e+03 + 8.446e+03 +	6.650e+03 + 4.389e+03 +	3.898e+03 + 3.898e+03 +	4.459e+03 + 4.459e+03 +	2.929e+03 + 2.929e+03 +	2.941e+03 + 2.941e+03 +	2.929e+03 + 2.929e+03 +	2.929e+03 + 2.929e+03 +	2.929e+03 + 2.929e+03 +	2.929e+03
	Std	1.884e+02 6.716e+00	2.121e+01 9.030e+02	4.928e+02 7.997e+02	2.128e+02 5.960e+02	8.582e+00 8.582e+00	2.364e+01 2.364e+01	6.908e+00 6.908e+00	1.714e+01 1.714e+01	1.714e+01 1.714e+01	1.714e+01 1.714e+01	1.714e+01
+/-/≈/-:		10/0/0	6/3/1	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	-
Avg. ranks:		7.2	26	5.9	120	106	93	82	97	30	5.0	1.8

Table 4 Experimental results and statistical analysis on 50-D CEC2020

Func.	COA	MPA	AGTO	AOA	SOA	ZOA	GIO	EVO	IMPA	EMPA	CMPA	MEMPA
F1	Mean	3.799e+10 + 3.018e+06 +	2.717e+08 +	1.169e+11 +	1.065e+11 +	6.414e+10 +	6.089e+10 +	6.934e+10 +	4.412e+06 +	1.750e+07 +	4.330e+06 +	3.324e+05
	Std	9.233e+09	2.199e+06	9.230e+08	5.100e+09	6.715e+09	6.941e+09	6.352e+09	9.721e+09	1.912e+06	8.729e+06	2.416e+06
F2	Mean	3.924e+12 + 3.762e+08 +	9.239e+09 +	1.388e+13 +	1.195e+13 +	7.134e+12 +	6.161e+12 +	7.438e+12 +	6.404e+08 +	1.876e+09 +	6.782e+08 +	3.231e+07
	Std	9.483e+11	1.724e+08	2.065e+10	1.052e+12	8.462e+11	9.485e+11	6.724e+11	1.374e+12	6.021e+08	8.812e+08	4.989e+08
F3	Mean	1.137e+12 + 1.211e+08 +	2.829e+09 +	4.783e+12 +	4.032e+12 +	2.310e+12 +	1.935e+12 +	2.517e+12 +	1.614e+08 +	5.480e+08 +	1.899e+08 +	8.005e+06
	Std	2.867e+11	6.602e+07	7.289e+09	3.980e+11	3.202e+11	2.375e+11	1.800e+11	3.345e+11	7.530e+07	2.661e+08	8.273e+07
F4	Mean	7.301e+04 + 1.945e+03 +	2.359e+03 +	5.745e+03 +	2.084e+06 +	6.024e+05 +	1.623e+05 +	4.835e+05 +	1.941e+03 +	1.964e+03 +	1.943e+03 +	1.929e+03
	Std	5.645e+04	1.481e+01	4.285e+02	1.985e+06	5.917e+05	2.596e+05	8.295e+04	2.815e+05	1.048e+01	1.524e+01	1.117e+01
F5	Mean	5.175e+07 + 4.096e+05 +	9.896e+05 +	1.275e+09 +	3.542e+08 +	1.381e+08 +	6.663e+07 +	7.689e+07 +	4.683e+05 +	7.112e+05 +	4.431e+05 +	2.962e+05
	Std	1.456e+07	1.604e+05	9.070e+05	5.354e+08	1.212e+08	9.578e+07	2.351e+07	5.924e+07	1.774e+05	3.310e+05	1.533e+05
F6	Mean	4.041e+07 + 1.646e+04 +	1.168e+04 +	2.181e+10 +	6.884e+07 +	1.085e+09 +	6.976e+08 +	1.243e+09 +	1.661e+04 +	3.203e+04 +	1.761e+04 +	7.126e+03
	Std	2.506e+07	7.947e+03	5.705e+03	7.857e+09	2.072e+07	1.046e+09	4.359e+08	1.272e+09	7.785e+03	1.465e+04	8.108e+03
F7	Mean	2.626e+08 + 1.136e+06 +	2.055e+06 +	3.165e+10 +	1.603e+10 +	1.620e+09 +	8.679e+08 +	1.518e+09 +	1.430e+06 +	2.562e+06 +	1.491e+06 +	7.334e+05
	Std	9.800e+07	6.565e+05	3.396e+06	1.086e+10	5.394e+09	9.411e+08	4.136e+08	1.586e+09	6.979e+05	1.832e+06	9.275e+05
F8	Mean	2.914e+03 + 2.478e+03 –	3.132e+03 +	1.572e+04 +	9.882e+03 +	7.309e+03 +	4.322e+03 +	4.114e+03 +	2.481e+03 –	2.487e+03 ≈	2.480e+03 –	2.496e+03
	Std	1.549e+02	1.698e+01	1.215e+03	1.275e+03	7.255e+02	1.727e+03	4.425e+02	7.712e+02	1.685e+01	1.292e+01	1.813e+01
F9	Mean	3.778e+04 + 2.720e+03 +	8.279e+03 +	6.883e+04 +	6.329e+04 +	5.018e+04 +	4.118e+04 +	5.308e+04 +	2.781e+03 +	3.063e+03 +	2.744e+03 +	2.661e+03
	Std	7.033e+03	2.009e+01	4.817e+03	1.951e+03	1.313e+03	4.582e+03	4.423e+03	6.949e+03	1.758e+02	4.130e+02	3.249e+01
F10	Mean	7.855e+03 + 3.400e+03 ≈	3.617e+03 +	3.301e+04 +	2.104e+04 +	1.297e+04 +	9.484e+03 +	1.413e+04 +	3.462e+03 +	3.457e+03 +	3.410e+03 +	3.366e+03
	Std	1.038e+03	1.086e+02	2.092e+02	4.209e+03	2.081e+03	1.617e+03	1.150e+03	3.212e+03	1.374e+02	1.335e+02	1.054e+02
+/-:	10/0/0	8/1/1	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	9/0/1	9/1/0	-	-	-
Avg. ranks:	6.9	2.2	5.6	12.0	10.7	9.5	8.2	9.6	3.5	5.0	3.4	1.4

Table 5 Experimental results and statistical analysis on 100-D CEC2020

Func.	COA	MPA	AGTO	AOA	SOA	ZOA	GIO	EVO	IMPA	EMPA	CMPA	MEMPA
F1	Mean	1.427e+11 + 3.851e+08 + 6.889e+09 + 2.763e+11 + 2.437e+11 + 1.949e+11 + 1.823e+11 + 2.047e+11 + 6.365e+08 + 2.232e+09 + 5.935e+08 + 3.423e+07										
	Std	1.858e+10 - 1.426e+08	3.736e+09	7.321e+09	3.316e+09	1.284e+10	7.824e+09	2.039e+10	2.025e+08	7.674e+08	1.606e+08	1.358e+07
F2	Mean	1.548e+13 + 3.928e+10 + 1.095e+12 + 3.283e+13 + 2.799e+13 + 2.212e+13 + 1.990e+13 + 2.258e+13 + 7.169e+10 + 1.916e+11 + 5.696e+10 + 3.956e+09										
	Std	2.436e+12 - 1.150e+10	5.928e+11	8.647e+11	3.165e+11	1.328e+12	1.040e+12	2.205e+12	2.469e+10	6.919e+10	1.694e+10	2.56e+09
F3	Mean	5.439e+12 + 1.446e+10 + 3.427e+11 + 1.120e+13 + 8.867e+12 + 7.296e+12 + 6.312e+12 + 7.855e+12 + 2.300e+10 + 7.548e+10 + 2.437e+10 + 1.489e+09										
	Std	5.839e+11 - 3.066e+09	2.505e+11	4.209e+11	8.809e+10	5.095e+11	3.850e+11	9.630e+11	5.621e+09	2.469e+10	7.737e+09	8.433e+08
F4	Mean	6.379e+05 + 2.142e+03 + 3.877e+03 + 8.738e+06 + 5.735e+06 + 2.078e+06 + 9.452e+05 + 2.292e+06 + 2.173e+03 + 2.405e+03 + 2.167e+03 + 2.036e+03										
	Std	2.398e+05 - 5.650e+01	1.356e+03	1.657e+06	8.806e+05	6.142e+05	2.582e+05	1.024e+06	6.172e+01	1.763e+02	5.622e+01	2.85e+01
F5	Mean	3.414e+08 + 1.092e+07 + 1.606e+07 + 3.485e+09 + 8.586e+08 + 5.525e+08 + 3.500e+08 + 3.807e+08 + 1.309e+07 + 2.379e+07 + 9.994e+06 + 5.116e+06										
	Std	5.936e+07 - 3.194e+06	6.581e+06	7.848e+08	1.407e+08	1.719e+08	6.533e+07	1.584e+08	4.050e+06	7.134e+06	4.094e+06	1.998e+06
F6	Mean	2.354e+09 + 2.999e+05 + 3.100e+05 ≈ 8.750e+10 + 4.683e+10 + 1.179e+10 + 5.251e+09 + 8.415e+09 + 2.829e+05 + 4.475e+05 + 3.085e+05 + 1.159e+05										
	Std	1.503e+09 - 2.985e+05	5.158e+05	1.651e+10	1.411e+10	4.320e+09	2.032e+09	4.176e+09	2.663e+05	5.119e+05	3.455e+05	9.763e+04
F7	Mean	2.247e+09 + 1.489e+07 + 2.378e+07 + 4.991e+10 + 2.584e+10 + 9.238e+09 + 4.151e+09 + 8.558e+09 + 1.658e+07 + 3.381e+07 + 2.072e+07 + 7.266e+06										
	Std	6.662e+08 - 7.075e+06	1.461e+07	5.924e+09	3.793e+09	3.324e+09	1.287e+09	3.961e+09	8.795e+06	2.049e+07	8.887e+06	4.482e+06
F8	Mean	4.600e+03 + 2.621e+03 + 5.008e+03 + 3.151e+04 + 2.362e+04 + 1.834e+04 + 1.079e+04 + 1.068e+04 + 2.625e+03 + 2.675e+03 + 2.633e+03 + 2.559e+03										
	Std	6.368e+02 - 3.801e+01	3.555e+03	1.765e+03	9.060e+02	3.921e+03	1.078e+03	2.172e+03	3.568e+01	4.387e+01	3.246e+01	2.935e+01
F9	Mean	1.356e+05 + 7.612e+03 + 3.768e+04 + 1.870e+05 + 1.616e+05 + 1.511e+05 + 1.464e+05 + 1.664e+05 + 9.520e+03 + 1.739e+04 + 9.145e+03 + 3.754e+03										
	Std	9.121e+03 - 1.398e+03	2.213e+04	6.069e+03	8.556e+02	6.231e+03	3.316e+03	1.182e+04	2.170e+03	5.610e+03	1.693e+03	9.001e+02
F10	Mean	1.440e+04 + 3.873e+03 + 4.528e+03 + 5.008e+04 + 4.197e+04 + 2.271e+04 + 1.648e+04 + 4.208e+03 + 3.882e+03 + 4.173e+03 + 3.897e+03 + 3.623e+03										
	Std	1.460e+03 - 1.308e+02	3.803e+02	6.174e+03	4.654e+03	2.631e+03	1.320e+03	4.208e+03	1.595e+02	2.088e+02	1.514e+02	1.158e+02
++/-:		10/0/0	10/0/0	9/1/0	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	-
Avg. ranks:		6.9	2.2	5.8	12.0	10.9	9.4	8.1	9.6	3.4	5.3	34

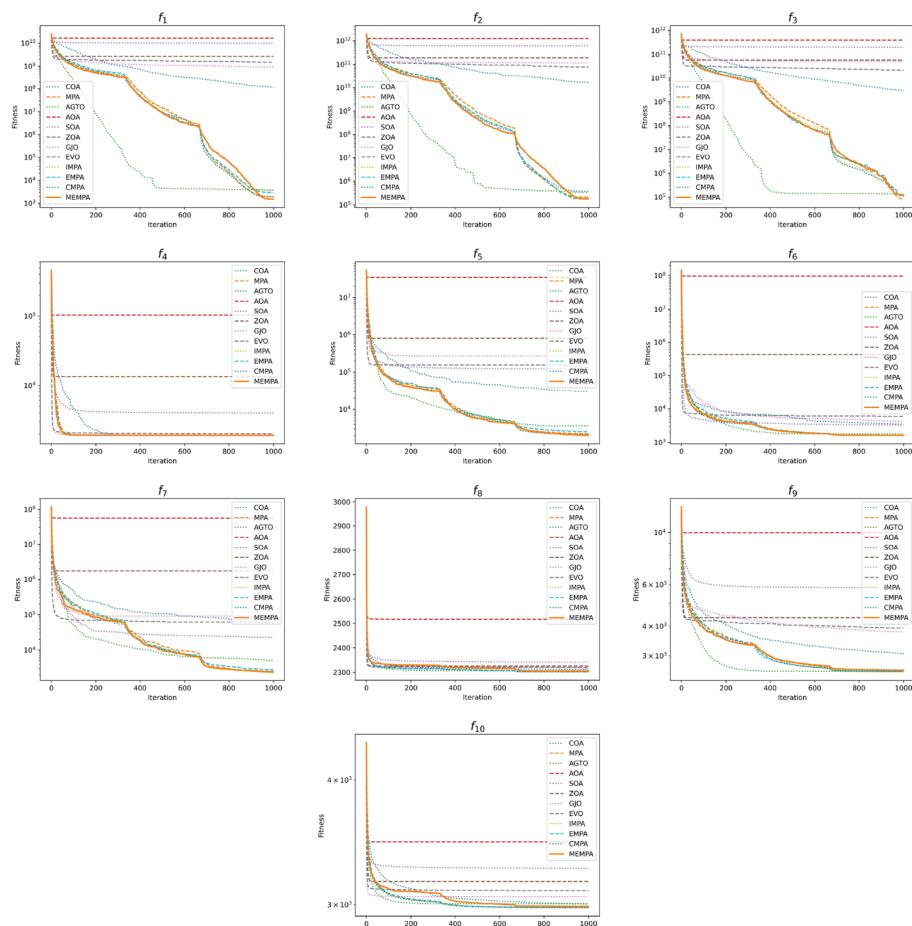


Fig. 5 Convergence curves of optimizers in 10-D CEC2020

an impressive average ranking of 2.9 and was considered the best-performing optimizer. When the dimension rises to 30-D, the Sobol sequence-based initialization and the mutualism mechanism cooperatively accelerate the optimization of MPA and allow MPA-ss+mm to achieve the best performance among eight variants with the rank of 2.5. In both 50-D and 100-D instances, MEMPA showed excellent performance with average rankings of 2.3 and 1.7, respectively, and we attribute these improvements to the intelligent integration of three strategies, where the Sobol sequence-based initialization allows MEMPA to generate a more uniform population within the search domain, the mutualism mechanism enhances the exploitative searchability of MEMPA, and the distance-based selection mechanism maintains the population diversity even in the late stage of optimization.

In summary, the combination of these strategies leads to a notable improvement in optimization convergence across a range of problem dimensions. This highlights the importance of synergistically applying these three strategies to achieve optimal performance. We recommend utilizing them together to maximize the efficiency and effectiveness of MEMPA.

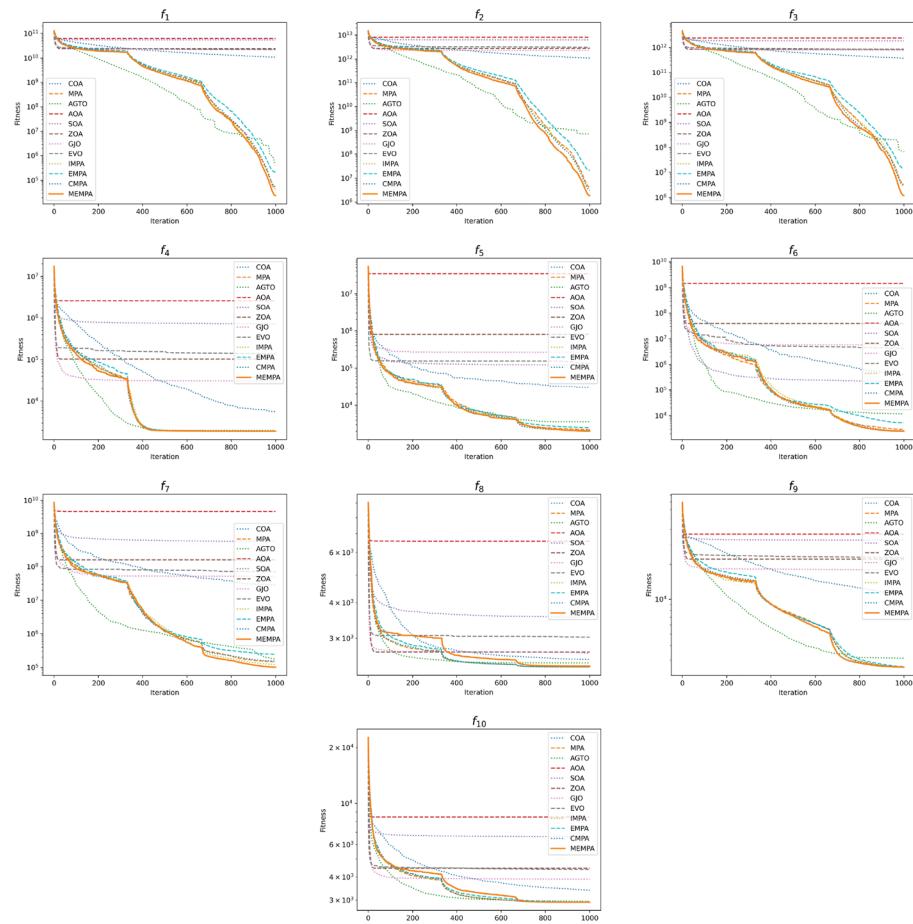


Fig. 6 Convergence curves of optimizers in 30-D CEC2020

Experimental results on engineering problems

To investigate the performance of MEMPA in practical engineering problems, this section applies the proposed MEMPA to six constrained engineering optimization problems in the field of structure: Three-bar Truss Design Problem, Pressure Vessel Design Problem, Step-cone Pulley Problem, Robot Gripper Problem, Speed Reducer Design Problem, and Gas Transmission Compressor Design Problem. The population size and the maximum number of iterations of optimizers are fixed at 30 and 1000, respectively. All optimizers are independently repeated 30 times, and the best fitness values are in bold.

Three-bar truss design problem

The Three-bar Truss Design Problem is a constrained optimization problem used in building structures to provide support and stability for structures such as Bridges, roofs, and towers. The problem has two optimization variables and three constraints, and the goal is to minimize the weight of the three-bar truss. A demonstration of the Three-bar Truss Design Problem is presented in Fig. 9, and the numerical results are summarized in Table 10. Figure 10 presents the convergence curves and boxplots of optimizers.

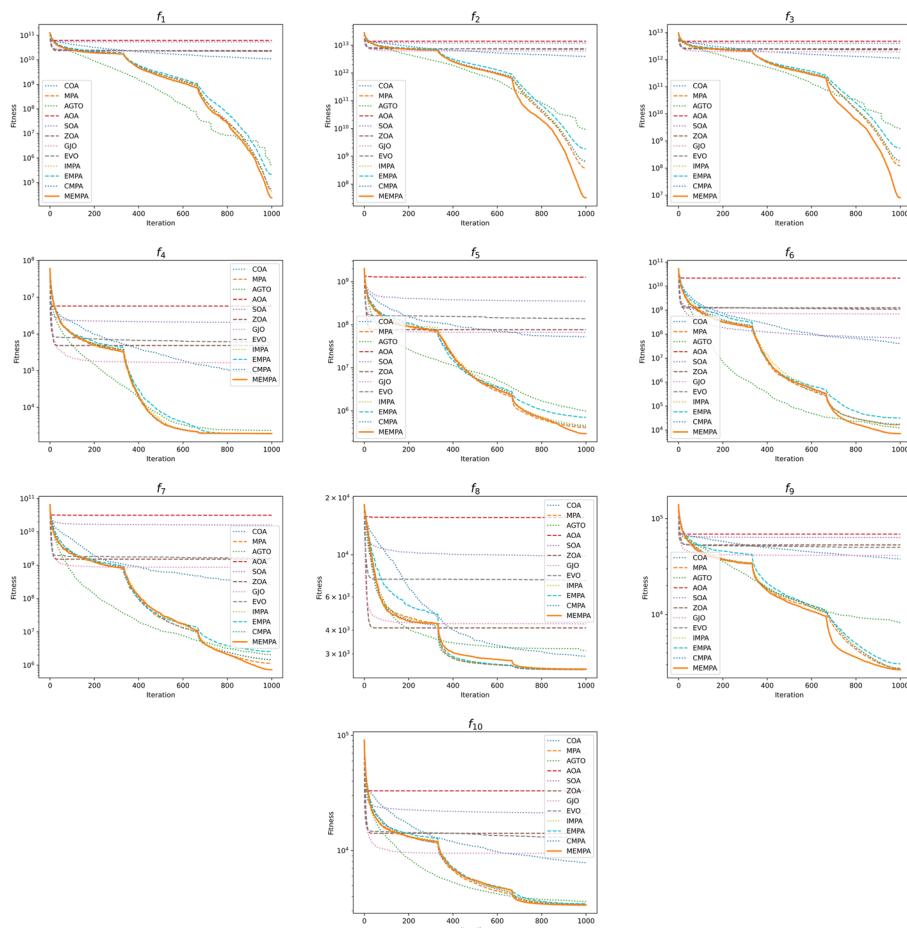


Fig. 7 Convergence curves of optimizers in 50-D CEC2020

In the Three-bar Truss Design Problem, MEMPA is ranked as the best optimizer, while the improved variants of MPA are ranked subsequently. Although the statistical significance between MEMPA and other MPA-based optimizers does not exist, the performance of MEMPA significantly outperforms other optimizers. We attribute the remarkable performance of MEMPA to the inherited architecture from MPA, which retains the core strengths of the original algorithm.

Pressure vessel design problem

The Pressure Vessel Design Problem is a classical constrained optimization problem in physical and chemical engineering, which has four optimization variables and four constraints with the goal of minimizing the total cost of welding, materials, and manufacturing. The manufacturing configuration for this problem is presented in Fig. 11, and the experimental results are summarized in Table 11. Convergence curves and boxplots are presented in Fig. 12.

In the Pressure Vessel Design Problem, MEMPA, CMPA, and IMPA achieve the top three performances, with MEMPA ranking first, closely followed by CMPA, and IMPA in third. Although MEMPA performs similarly to MPA and its advanced variants with no statistically significant difference, it still demonstrates a clear numerical advantage.

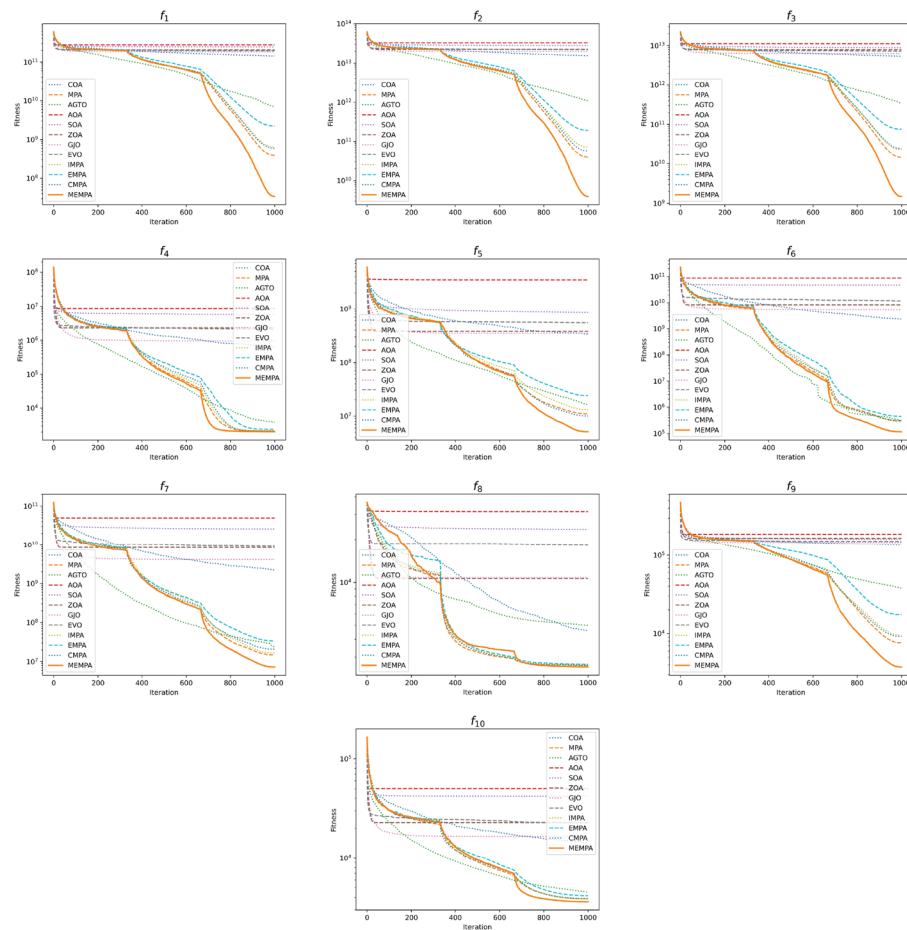


Fig. 8 Convergence curves of optimizers in 100-D CEC2020

This performance is attributed to the integration of the three proposed strategies within MEMPA, which collectively accelerate optimization convergence.

Step-cone pulley problem

The Step-Cone Pulley Problem is a classical constraint optimization problem where the main goal is to minimize the weight of a 4-step cone pulley using five variables, four of which are the diameter of each Step of the pulley and the last variable is the width of the pulley, for a total of 11 constraints. Figure 13 presents a demonstration of the step cone pulley, Table 12 summarizes the numerical results, and convergence curves and boxplots are presented in Fig. 14.

In the Step-cone Pulley Problem, AGTO, MEMPA, and MPA rank as the top three optimizers, with AGTO achieving the best performance, followed by MEMPA and MPA. Although MEMPA performs significantly worse than AGTO, its performance is approximately equal to MPA and significantly better than other competitor algorithms. Despite not securing the top position, these results highlight the competitiveness and effectiveness of MEMPA in solving the Step-cone Pulley Problem.

Table 6 Results of ablation experiments in 10-D CEC2020

Func		MPA	MPA-ss	MPA-mm	MPA-dsm	MPA-ss+mm	MPA-ss+dsm	MPA-mm+dsm	MEMPA
F1	Mean	1.955e+03	2.045e+03	2.317e+03	1.746e+03	1.380e+03	2.474e+03	2.403e+03	1.523e+03
	Std	1.941e+03	2.596e+03	2.624e+03	2.513e+03	1.233e+03	2.766e+03	2.684e+03	1.532e+03
F2	Mean	2.073e+05	3.084e+05	2.886e+05	3.257e+05	1.219e+05	3.243e+05	2.131e+05	1.777e+05
	Std	2.519e+05	3.490e+05	3.543e+05	3.210e+05	1.511e+05	3.423e+05	2.810e+05	1.895e+05
F3	Mean	8.204e+04	9.581e+04	1.083e+05	1.044e+05	1.520e+05	1.156e+05	1.196e+05	1.215e+05
	Std	1.008e+05	1.150e+05	1.365e+05	1.393e+05	1.730e+05	1.314e+05	1.075e+05	1.455e+05
F4	Mean	1.901e+03	1.901e+03	1.901e+03	1.901e+03	1.901e+03	1.902e+03	1.901e+03	1.901e+03
	Std	4.002e-01	3.387e-01	3.523e-01	7.772e-01	3.303e-01	6.667e-01	7.682e-01	5.637e-01
F5	Mean	2.259e+03	2.043e+03	2.106e+03	2.141e+03	2.147e+03	2.033e+03	2.193e+03	2.023e+03
	Std	3.625e+02	2.691e+02	2.927e+02	2.952e+02	3.950e+02	2.521e+02	3.560e+02	2.434e+02
F6	Mean	1.610e+03	1.607e+03	1.606e+03	1.608e+03	1.608e+03	1.608e+03	1.610e+03	1.621e+03
	Std	1.831e+01	1.016e+01	1.544e+01	1.043e+01	1.320e+01	1.061e+01	1.350e+01	7.018e+01
F7	Mean	2.528e+03	2.382e+03	2.422e+03	2.538e+03	2.463e+03	2.438e+03	2.500e+03	2.389e+03
	Std	2.750e+02	1.774e+02	2.118e+02	3.103e+02	2.535e+02	3.140e+02	3.158e+02	2.203e+02
F8	Mean	2.301e+03	2.302e+03	2.302e+03	2.305e+03	2.301e+03	2.304e+03	2.304e+03	2.305e+03
	Std	1.133e+00	1.200e+00	1.305e+00	2.361e+00	9.580e-01	1.646e+00	1.562e+00	2.517e+00
F9	Mean	2.572e+03	2.568e+03	2.567e+03	2.584e+03	2.568e+03	2.586e+03	2.592e+03	2.610e+03
	Std	5.674e+01	5.585e+01	4.713e+01	8.435e+01	6.314e+01	6.329e+01	8.378e+01	8.346e+01
F10	Mean	2.979e+03	2.978e+03	2.979e+03	2.988e+03	2.978e+03	2.992e+03	2.985e+03	2.989e+03
	Std	1.526e+00	1.345e+00	5.263e+00	1.132e+01	1.439e+00	1.233e+01	9.513e+00	1.246e+01
+/-/-:	-	0/8/2	0/9/1	2/8/0	0/10/0	3/6/1	2/8/0	2/6/2	
Avg. ranks:	4.0	2.9	3.3	5.8	3.4	5.6	6.1	4.9	

Bold indicates best value

Robot gripper problem

Robotic Gripper Design is a complex problem in mechanical structural engineering, and the demonstration is shown in Figure 12. This problem uses the difference between the minimum force generated and the maximum force as the objective function. It contains 7 design variables and 6 nonlinear design constraints related to the robot. Figure 15 demonstrates the Robot Gripper Problem, while the numerical results of MPA and other competitors are summarized in Table 13, and convergence curves and boxplots are presented in Fig. 16.

In the Robot Gripper Problem, the top three optimizers are MEMPA, CMPA, and AGTO, with MEMPA significantly outperforming all competitor algorithms. These results underscore the remarkable competitiveness of MEMPA in this optimization task. Additionally, algorithms like EVO may find infeasible solutions at the end of optimization, a challenge observed not only in the Robot Gripper Problem but also in other instances. In contrast, MEMPA consistently finds feasible solutions across every independent trial, further demonstrating its robustness and reliability.

Table 7 Results of ablation experiments in 30-D CEC2020

Func		MPA	MPA-ss	MPA-mm	MPA-dsm	MPA-ss+mm	MPA-ss+dsm	MPA-mm+dsm	MEMPA
F1	Mean	4.511e+04	3.243e+04	2.184e+04	4.513e+04	7.943e+03	4.433e+04	7.388e+03	2.390e+04
	Std	2.907e+04	2.190e+04	1.730e+04	3.632e+04	6.656e+03	3.658e+04	4.648e+03	1.974e+04
F2	Mean	4.218e+06	2.183e+06	1.899e+06	2.350e+06	6.571e+05	2.568e+06	1.063e+06	1.870e+06
	Std	3.277e+06	1.535e+06	1.482e+06	1.656e+06	6.052e+05	2.958e+06	9.788e+05	2.454e+06
F3	Mean	3.310e+06	1.887e+06	1.244e+06	1.451e+06	2.603e+05	1.748e+06	3.419e+05	1.181e+06
	Std	2.219e+06	1.005e+06	7.478e+05	9.786e+05	2.554e+05	1.034e+06	2.331e+05	6.255e+05
F4	Mean	1.908e+03	1.908e+03	1.907e+03	1.909e+03	1.907e+03	1.911e+03	1.909e+03	1.909e+03
	Std	2.254e+00	2.180e+00	1.620e+00	2.818e+00	2.131e+00	2.464e+00	2.645e+00	2.319e+00
F5	Mean	7.668e+04	7.498e+04	7.114e+04	6.531e+04	7.291e+04	7.588e+04	6.222e+04	7.555e+04
	Std	1.978e+04	2.475e+04	2.634e+04	2.225e+04	1.906e+04	1.976e+04	1.438e+04	3.128e+04
F6	Mean	2.979e+03	2.887e+03	2.434e+03	2.707e+03	2.549e+03	2.547e+03	3.078e+03	2.510e+03
	Std	1.436e+03	1.613e+03	1.416e+03	1.050e+03	1.267e+03	9.586e+02	2.771e+03	7.940e+02
F7	Mean	1.476e+05	1.040e+05	1.164e+05	1.231e+05	7.857e+04	1.237e+05	9.574e+04	1.019e+05
	Std	9.516e+04	6.410e+04	7.717e+04	8.232e+04	5.201e+04	1.038e+05	4.987e+04	5.613e+04
F8	Mean	2.379e+03	2.381e+03	2.375e+03	2.396e+03	2.374e+03	2.396e+03	2.393e+03	2.397e+03
	Std	5.733e+00	7.398e+00	5.099e+00	1.345e+01	5.139e+00	1.337e+01	1.053e+01	1.319e+01
F9	Mean	2.613e+03	2.610e+03	2.607e+03	2.609e+03	2.610e+03	2.609e+03	2.602e+03	2.607e+03
	Std	3.178e+00	3.271e+00	3.531e+00	3.055e+00	4.303e+01	3.590e+00	1.377e+00	3.288e+00
F10	Mean	2.928e+03	2.927e+03	2.928e+03	2.923e+03	2.924e+03	2.923e+03	2.923e+03	2.926e+03
	Std	6.716e+00	5.826e+00	4.911e+00	2.768e+00	2.874e+00	2.564e+00	4.384e+00	1.714e+01
+/-	-	0/6/4	0/4/6	1/4/5	0/2/8	2/4/4	1/2/7	1/3/6	
Avg. ranks:		6.8	5.2	3.4	5.2	2.5	5.6	3.1	4.2

Bold indicates best value

Speed reducer design problem

Speed Reducer Design Problem is an important constraint optimization problem in mechanical braking systems, as shown in Fig. 17. The problem mainly involves 7 variables and 11 constraints, and the goal is to minimize the weight of the reducer. Table 14 summarizes the numerical results, while convergence curves and boxplots of MEMPA and competitor algorithms are presented in Fig. 18.

In the Speed Reducer Design Problem, MPA, MEMPA, and CMPA rank as the top-performing optimizers, where the performance of MEMPA is approximate to MPA and significantly outperforms other competitor algorithms. This remarkable performance underscores the competitiveness of MEMPA in addressing complex engineering design challenges. The ability of MEMPA to consistently achieve high-quality solutions demonstrates its effectiveness, further validating the integration of the three proposed strategies, which collectively enhance optimization convergence and solution accuracy in comparison to competitors.

Table 8 Results of ablation experiments in 50-D CEC2020

Func		MPA	MPA-ss	MPA-mm	MPA-dsm	MPA-ss+mm	MPA-ss+dsm	MPA-mm+dsm	MEMPA
F1	Mean	3.018e+06	2.822e+06	1.683e+06	2.994e+06	2.446e+05	3.135e+06	2.947e+05	3.324e+05
	Std	2.199e+06	1.541e+06	1.014e+06	1.454e+06	1.557e+05	1.966e+06	1.793e+05	1.977e+05
F2	Mean	3.762e+08	3.461e+08	2.839e+08	4.712e+08	4.562e+07	3.574e+08	3.200e+07	3.231e+07
	Std	1.724e+08	1.637e+08	3.489e+08	4.976e+08	8.617e+07	1.872e+08	2.856e+07	7.518e+07
F3	Mean	1.211e+08	1.138e+08	5.184e+07	1.134e+08	9.061e+06	1.135e+08	8.514e+06	8.005e+06
	Std	6.602e+07	5.705e+07	3.198e+07	5.543e+07	6.090e+06	4.832e+07	6.133e+06	9.763e+06
F4	Mean	1.945e+03	1.941e+03	1.939e+03	1.933e+03	1.934e+03	1.933e+03	1.930e+03	1.929e+03
	Std	1.481e+01	1.176e+01	1.000e+01	7.496e+00	1.443e+01	9.261e+00	8.621e+00	8.754e+00
F5	Mean	4.096e+05	4.281e+05	3.664e+05	4.863e+05	3.197e+05	4.480e+05	3.041e+05	2.962e+05
	Std	1.604e+05	1.432e+05	1.513e+05	1.788e+05	1.094e+05	1.704e+05	1.227e+05	8.710e+04
F6	Mean	1.646e+04	1.377e+04	1.336e+04	1.613e+04	7.128e+03	1.524e+04	8.135e+03	7.126e+03
	Std	7.947e+03	5.960e+03	6.967e+03	7.220e+03	2.575e+03	6.229e+03	3.630e+03	3.076e+03
F7	Mean	1.136e+06	1.481e+06	1.202e+06	1.253e+06	6.296e+05	1.644e+06	8.030e+05	7.334e+05
	Std	6.565e+05	8.991e+05	7.209e+05	6.849e+05	5.430e+05	1.197e+06	4.561e+05	6.487e+05
F8	Mean	2.478e+03	2.469e+03	2.473e+03	2.515e+03	2.464e+03	2.514e+03	2.490e+03	2.496e+03
	Std	1.698e+01	1.515e+01	1.592e+01	2.425e+01	1.497e+01	2.909e+01	2.616e+01	2.684e+01
F9	Mean	2.720e+03	2.723e+03	2.713e+03	2.698e+03	2.631e+03	2.719e+03	2.650e+03	2.661e+03
	Std	2.009e+01	1.924e+01	1.743e+02	1.847e+01	1.173e+01	1.316e+02	1.212e+02	1.400e+02
F10	Mean	3.400e+03	3.393e+03	3.384e+03	3.356e+03	3.387e+03	3.344e+03	3.369e+03	3.366e+03
	Std	1.086e+02	9.934e+01	1.105e+02	9.389e+01	9.925e+01	6.710e+01	9.759e+01	8.800e+01
+/-/≈:	-	0/9/1	0/6/4	1/7/2	0/1/9	1/7/2	0/2/8	1/1/8	
Avg. ranks:		6.6	5.9	4.4	5.7	2.6	5.9	2.6	2.3

Bold indicates best value

Gas transmission compressor design problem

Gas Transmission Compressor Design Problem is a well-known optimization problem that determines the minimum cost for a gas pipeline transmission system per day. A demonstration of the Gas Transmission Compressor Design Problem is in Fig. 19 and the detailed numerical results are summarized in Table 15. Figure 20 presents the convergence curves and boxplots of optimizers.

In the Gas Transmission Compressor Design Problem, MPA, IMPA, and CMPA are the top three optimizers, while MEMPA is ranked as the fifth-best optimizer. Although the rank of MEMPA is not outstanding, the statistical significance between MEMPA and other competitive optimizers does not exist. These results show that MEMPA can effectively handle complex design optimization tasks. The fact that MEMPA consistently achieves better performance than most optimizers highlights its robustness and the effectiveness of its integrated strategy.

Table 9 Results of ablation experiments in 100-D CEC2020

Func		MPA	MPA-ss	MPA-mm	MPA-dsm	MPA-ss+mm	MPA-ss+dsm	MPA-mm+dsm	MEMPA
F1	Mean	3.851e+08	3.801e+08	1.871e+08	3.999e+08	3.451e+07	3.684e+08	3.317e+07	3.423e+07 –
	Std	1.426e+08	1.242e+08	7.298e+07	1.106e+08	1.487e+07	8.821e+07	1.515e+07	1.358e+07
F2	Mean	3.928e+10	3.733e+10	2.087e+10	4.133e+10	4.059e+09	4.211e+10	3.959e+09	3.956e+09 –
	Std	1.150e+10	8.332e+09	5.956e+09	1.004e+10	1.964e+09	1.135e+10	2.527e+09	2.526e+09
F3	Mean	1.446e+10	1.404e+10	6.358e+09	1.228e+10	1.823e+09	1.349e+10	1.490e+09	1.489e+09 –
	Std	3.066e+09	4.757e+09	2.083e+09	3.734e+09	1.174e+09	3.398e+09	5.994e+08	8.433e+08
F4	Mean	2.142e+03	2.145e+03	2.123e+03	2.084e+03	2.069e+03	2.105e+03	2.050e+03	2.036e+03 –
	Std	5.650e+01	5.175e+01	6.051e+01	4.054e+01	3.615e+01	4.192e+01	3.031e+01	2.895e+01
F5	Mean	1.092e+07	1.100e+07	7.826e+06	1.139e+07	5.452e+06	1.155e+07	5.828e+06	5.116e+06 –
	Std	3.194e+06	4.055e+06	2.963e+06	3.825e+06	1.751e+06	5.962e+06	2.661e+06	1.998e+06
F6	Mean	2.999e+05	2.823e+05	2.542e+05	2.539e+05	1.053e+05	1.931e+05	8.474e+04	1.159e+05 –
	Std	2.985e+05	3.125e+05	2.974e+05	2.295e+05	5.597e+04	1.489e+05	2.969e+04	9.763e+04
F7	Mean	1.489e+07	1.553e+07	1.311e+07	1.582e+07	6.379e+06	1.387e+07	6.630e+06	7.266e+06 –
	Std	7.075e+06	8.165e+06	6.989e+06	7.808e+06	3.315e+06	7.256e+06	4.899e+06	4.482e+06
F8	Mean	2.621e+03	2.615e+03	2.610e+03	2.616e+03	2.566e+03	2.603e+03	2.575e+03	2.559e+03 –
	Std	3.801e+01	2.806e+01	2.951e+01	4.781e+01	3.157e+01	2.876e+01	3.405e+01	2.935e+01
F9	Mean	7.612e+03	7.303e+03	5.532e+03	7.096e+03	3.672e+03	6.944e+03	3.573e+03	3.754e+03 –
	Std	1.398e+03	9.804e+02	1.419e+03	8.901e+02	7.592e+02	1.147e+03	7.347e+02	9.001e+02
F10	Mean	3.873e+03	3.831e+03	3.798e+03	3.694e+03	3.725e+03	3.698e+03	3.648e+03	3.623e+03 –
	Std	1.308e+02	1.302e+02	1.378e+02	1.240e+02	1.147e+02	1.138e+02	1.213e+02	1.158e+02
+/-	-	0/10/0	0/4/6	0/7/3	0/0/10	0/7/3	0/0/10	0/0/10	0/0/10
Avg. ranks:		7.1	6.6	4.7	6.0	2.6	5.4	1.9	1.7

Bold indicates best value

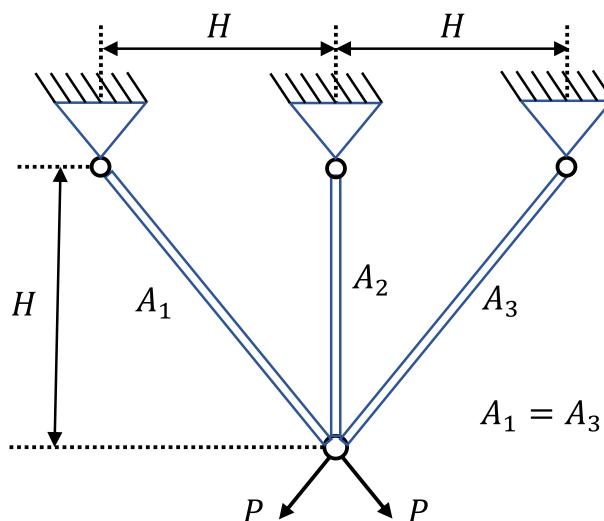
**Fig. 9** A demonstration of the Three-bar Truss Design Problem

Table 10 Experimental results and statistical analysis in the Three-bar Truss Design Problem

MAs	Mean	Std	Best	Worst
COA	2.639137e+02 +	1.188613e-02	2.638979e+02	2.639373e+02
MPA	2.638958e+02 ≈	1.558709e-06	2.638958e+02	2.638959e+02
AGTO	2.638960e+02 +	5.555954e-04	2.638958e+02	2.638986e+02
AOA	2.645087e+02 +	4.764017e-01	2.639014e+02	2.654941e+02
SOA	2.640210e+02 +	2.711202e-01	2.638960e+02	2.651876e+02
ZOA	2.641657e+02 +	4.018783e-01	2.638961e+02	2.652561e+02
GJO	2.723111e+02 +	5.589492e+00	2.645440e+02	2.828427e+02
EVO	2.685570e+02 +	3.023271e+00	2.644926e+02	2.757173e+02
IMPA	2.638958e+02 ≈	2.387991e-06	2.638958e+02	2.638959e+02
EMPA	2.638958e+02 ≈	2.865941e-06	2.638958e+02	2.638959e+02
CMPA	2.638958e+02 ≈	1.015391e-06	2.638958e+02	2.638958e+02
MEMPA	2.638958e+02	2.902841e-06	2.638958e+02	2.638958e+02

Bold indicates best value

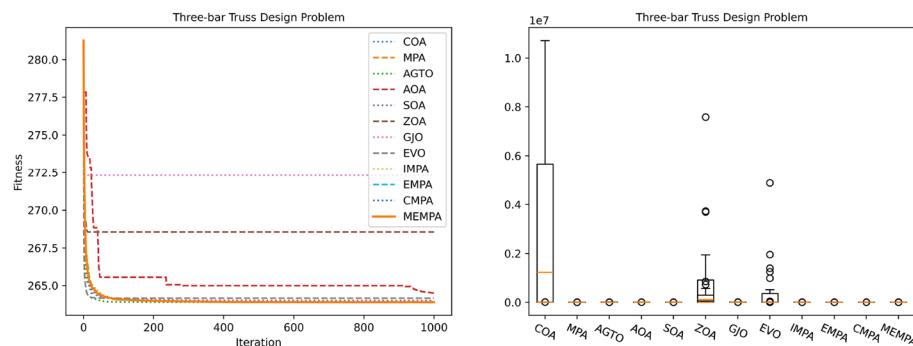


Fig. 10 Convergence curves and boxplots of optimizers in the Three-bar Truss Design Problem

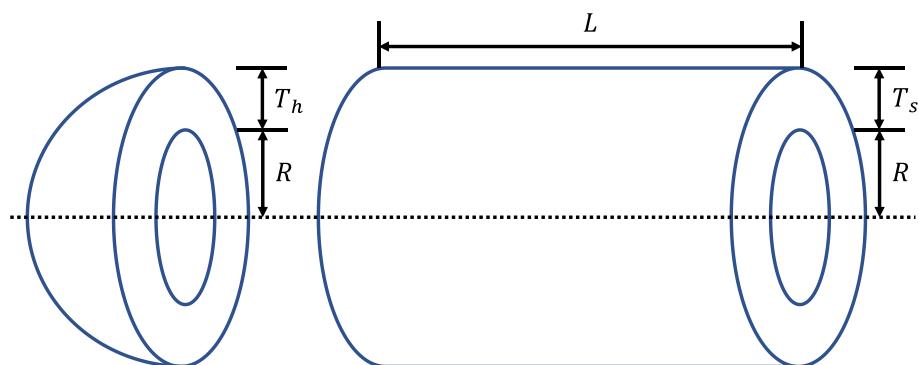


Fig. 11 A demonstration of the Pressure Vessel Design Problem

Experiments on intrusion detection

The intrusion detection system (IDS) is an important component for dynamically monitoring whether the target system has been invaded. IDS can be divided into misuse detection, anomaly detection, and specification-based detection according to

Table 11 Experimental results and statistical analysis in the pressure vessel design problem

MAs	Mean	Std	Best	Worst
COA	4.595724e+03 +	2.417403e+03	3.020116e+03	1.015480e+04
MPA	2.892440e+03 ≈	8.489068e-02	2.892423e+03	2.892897e+03
AGTO	4.706156e+03 +	2.565005e+03	2.892423e+03	8.333621e+03
AOA	7.804777e+03 +	2.707246e+03	2.968060e+03	1.078662e+04
SOA	2.898473e+03 +	3.782001e+00	2.893576e+03	2.911388e+03
ZOA	7.215500e+03 +	8.138932e+03	2.926013e+03	3.793678e+04
GJO	7.446183e+03 +	2.022993e+03	2.903058e+03	8.382053e+03
EVO	2.068638e+05 +	2.182776e+05	1.561037e+04	8.507140e+05
IMPA	2.892426e+03 ≈	5.309910e-03	2.892423e+03	2.892451e+03
EMPA	2.892467e+03 ≈	2.299289e-01	2.892423e+03	2.893705e+03
CMPA	2.892424e+03 ≈	1.346642e-03	2.892423e+03	2.892430e+03
MEMPA	2.892424e+03	1.456794e-03	2.892423e+03	2.892428e+03

Bold indicates best value

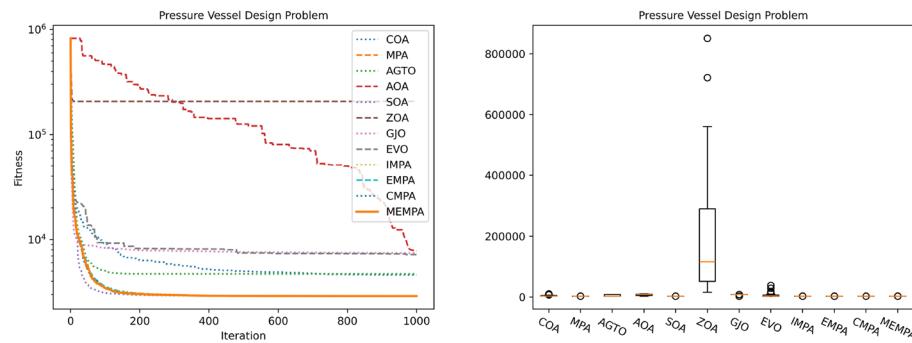
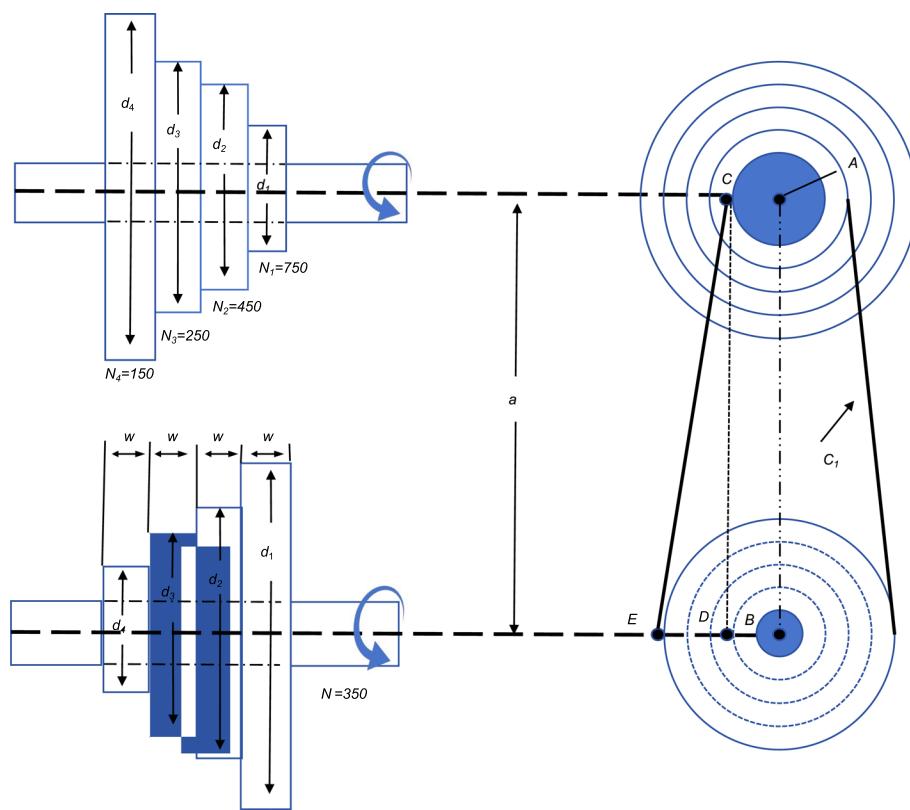


Fig. 12 Convergence curves and boxplots of optimizers in the Pressure Vessel Design Problem

different detection methods [41]. The network intrusion detection technology to be discussed in this section belongs to anomaly detection, which does not rely on the rule base of known attacks for detection. In the training phase, network traffic data is adopted to train the model. After training, the trained model is used as the main detection component to identify network behavior. The proposed MEMPA is used to optimize the weight and bias of ELM, and several independent MEMPA-ELM are integrated for intrusion detection. To maximize the diversity of the ensemble learning model, the training samples of all basic learners are randomly selected in proportion to the sample categories to reduce the data distribution imbalance. In the meantime, MEMPA-ELM and ELM are selected as the basic learner according to a certain probability, and data features with non-fixed dimensions are selected as the input of the basic learner. Finally, the majority voting method is used to combine these independent model outputs for final prediction. Figure 21 is the design framework of the integration model.

**Fig. 13** A demonstration of the step-cone pulley problem**Table 12** Experimental results and statistical analysis in the Step-cone Pulley Problem

MA	Mean	Std	Best	Worst
COA	1.665482e+06 +	2.630098e+06	1.621159e+05	1.558045e+07
MPA	1.917720e+01 ≈	1.110556e+00	1.749193e+01	2.171655e+01
AGTO	1.747781e+01 –	5.537076e-01	1.682059e+01	1.825599e+01
AOA	5.109824e+03 +	4.795596e+03	6.212194e+02	2.675572e+04
SOA	7.051986e+05 +	2.900635e+05	2.489315e+05	1.260118e+06
ZOA	9.482601e+05 +	1.147788e+06	3.019260e+01	4.771410e+06
GJO	1.579236e+04 +	6.768625e+03	4.453832e+03	2.944798e+04
EVO	3.555850e+06 +	4.385221e+06	1.552966e+05	2.399740e+07
IMPA	1.954468e+01 +	1.479550e+00	1.661189e+01	2.527916e+01
EMPA	1.930908e+01 ≈	1.238936e+00	1.696187e+01	2.245474e+01
CMPA	1.932290e+01 +	1.531766e+00	1.669563e+01	2.381525e+01
MEMPA	1.913546e+01	1.365459e+00	1.724809e+01	2.352712e+01

Bold indicates best value

Extreme learning machine (ELM)

Extreme Learning Machine (ELM) is an artificial neural network (ANN) primarily designed for regression, classification, and clustering tasks. Figure 22 demonstrates the architecture of ELM.

Unlike traditional neural networks, ELM stands out due to their extremely fast training process. The key idea behind ELM is to use a single hidden layer feedforward neural

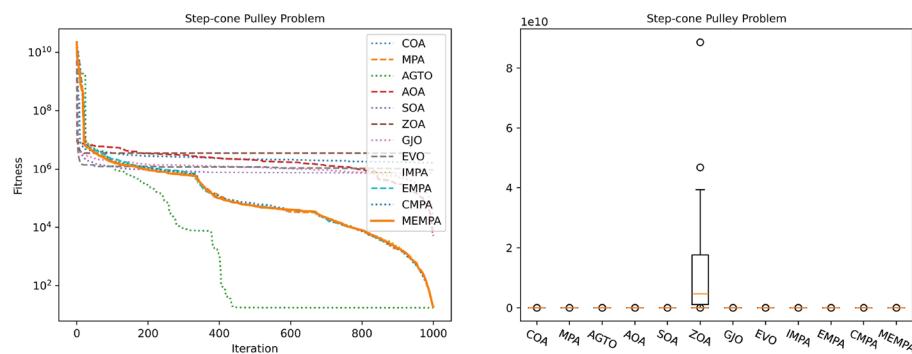


Fig. 14 Convergence curves and boxplots of optimizers in the step-cone pulley problem

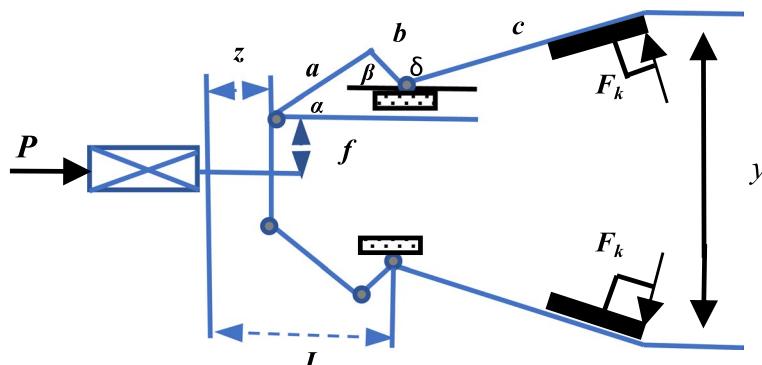


Fig. 15 A demonstration of the robot gripper problem

Table 13 Experimental results and statistical analysis in the robot gripper problem

MA	Mean	Std	Best	Worst
COA	6.597094e+00	2.380683e+00	4.016127e+00	1.433658e+01
MPA	3.268509e+00	1.069736e+00	2.526868e+00	5.196937e+00
AGTO	3.158066e+00	1.046776e+00	2.526826e+00	5.465808e+00
AOA	7.358481e+00	1.269607e+01	2.533022e+00	5.543771e+01
SOA	6.863608e+00	1.326859e+00	4.825734e+00	8.968443e+00
ZOA	6.982204e+00	1.529658e+00	4.712131e+00	1.167524e+01
GJO	4.269442e+00	7.683188e-02	3.916598e+00	4.289321e+00
EVO	1.246110e+10	1.847376e+10	1.108280e+01	8.855966e+10
IMPA	3.317857e+00	1.141414e+00	2.526838e+00	5.186516e+00
EMPA	3.158397e+00	9.783316e-01	2.526841e+00	4.994263e+00
CMPA	3.148375e+00	9.964837e-01	2.526835e+00	5.037645e+00
MEMPA	2.898025e+00	7.823980e-01	2.526933e+00	5.147200e+00

Bold indicates best value

network (SLFN) where the input weights and biases are randomly assigned and not updated during training. Only the output weights are optimized, typically using a least-squares solution. Mathematically, for an ELM with L hidden layers and N training samples (\mathbf{x}_i, t_i) , where $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T$ and $t_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T$, the predictive output \mathbf{o}_j is formulated using Eq. (13).

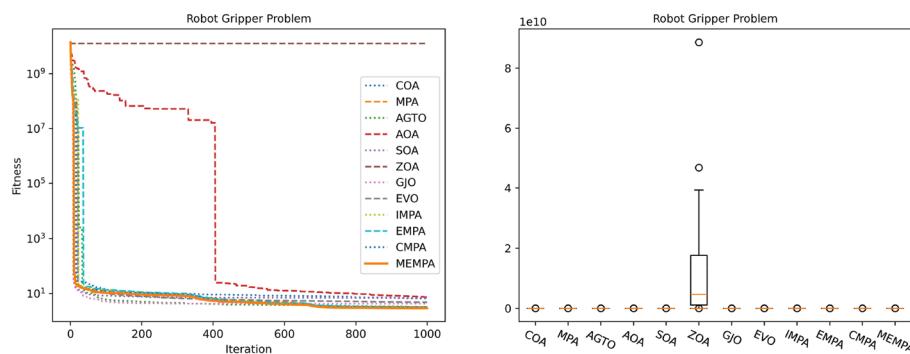


Fig. 16 Convergence curves and boxplots of optimizers in the robot gripper problem

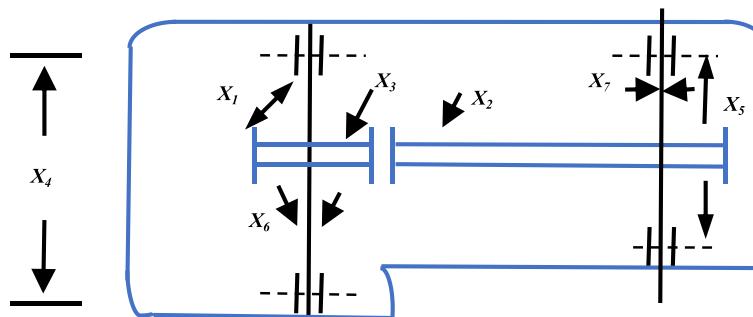


Fig. 17 A demonstration of the speed reducer design problem

Table 14 Experimental results and statistical analysis in the speed reducer design problem

MA	Mean	Std	Best	Worst
COA	2.934212e+06 +	3.363868e+06	3.101900e+03	1.070829e+07
MPA	2.987148e+03 ≈	4.470141e-01	2.986888e+03	2.988459e+03
AGTO	3.338634e+03 +	8.747907e+02	2.986884e+03	5.982998e+03
AOA	3.093149e+03 +	4.479377e+01	3.019441e+03	3.218614e+03
SOA	2.998707e+03 +	2.719028e+00	2.993909e+03	3.005666e+03
ZOA	4.117292e+05 +	9.602598e+05	3.108522e+03	4.879718e+06
GJO	3.012295e+03 +	1.280209e+01	2.999620e+03	3.052518e+03
EVO	7.873256e+05 +	1.604907e+06	3.050669e+03	7.580402e+06
IMPA	2.987350e+03 +	1.159925e+00	2.986889e+03	2.991720e+03
EMPA	2.987636e+03 +	2.162113e+00	2.986889e+03	2.995913e+03
CMPA	2.987474e+03 +	1.833245e+00	2.986890e+03	2.997022e+03
MEMPA	2.987160e+03	7.358111e-01	2.986888e+03	2.990831e+03

Bold indicates best value

$$\mathbf{o}_j = \sum_{i=1}^L \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) \quad (13)$$

where $g(x)$ is the activation function, $\mathbf{w}_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$ is the input weight, $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$ is the output weight, and b_i is the bias of the i^{th} hidden node.

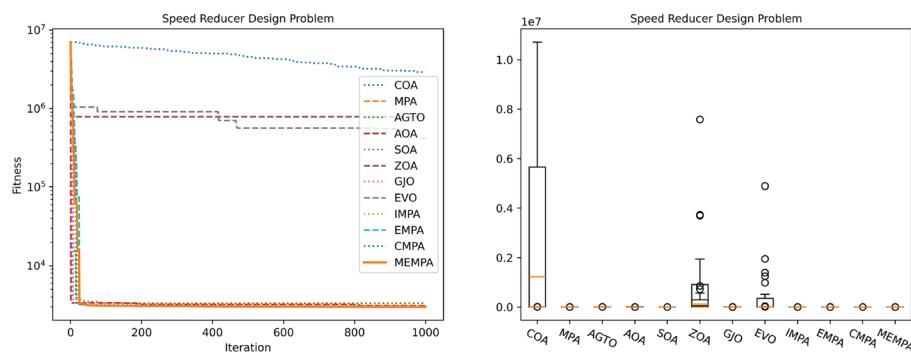


Fig. 18 Convergence curves and boxplots of optimizers in the speed reducer design problem

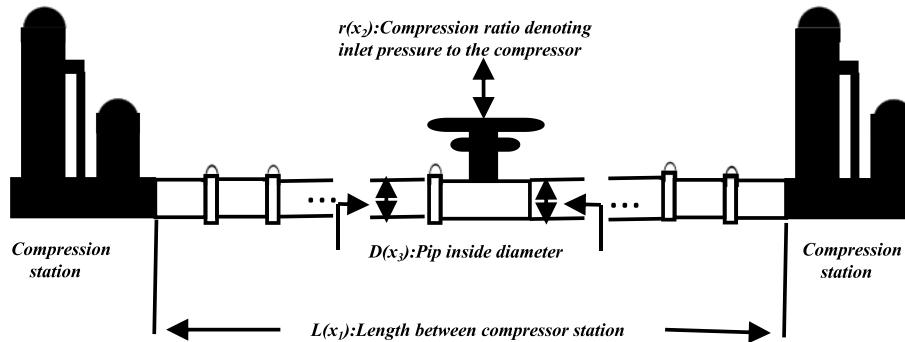


Fig. 19 A demonstration of the gas transmission compressor design problem

Table 15 Experimental results and statistical analysis in the gas transmission compressor design problem

MA	Mean	Std	Best	Worst
COA	3.016402e+06 +	1.883586e+04	2.976566e+06	3.052688e+06
MPA	2.964896e+06 ≈	1.743970e+00	2.964895e+06	2.964905e+06
AGTO	3.078103e+06 +	9.061033e+04	2.964895e+06	3.204728e+06
AOA	3.185812e+06 +	4.807741e+04	3.016531e+06	3.263044e+06
SOA	2.998329e+06 +	8.670342e+03	2.982665e+06	3.037849e+06
ZOA	3.027230e+06 +	2.838564e+04	2.975309e+06	3.077732e+06
GJO	2.971164e+06 +	2.704015e+03	2.967662e+06	2.979541e+06
EVO	7.401849e+06 +	1.320981e+06	4.739097e+06	1.033808e+07
IMPA	2.964896e+06 ≈	1.085307e+00	2.964895e+06	2.964901e+06
EMPA	2.964897e+06 ≈	1.488340e+00	2.964895e+06	2.964902e+06
CMPA	2.964896e+06 ≈	1.511520e+00	2.964895e+06	2.964903e+06
MEMPA	2.964898e+06	6.091436e+00	2.964895e+06	2.964929e+06

Bold indicates best value

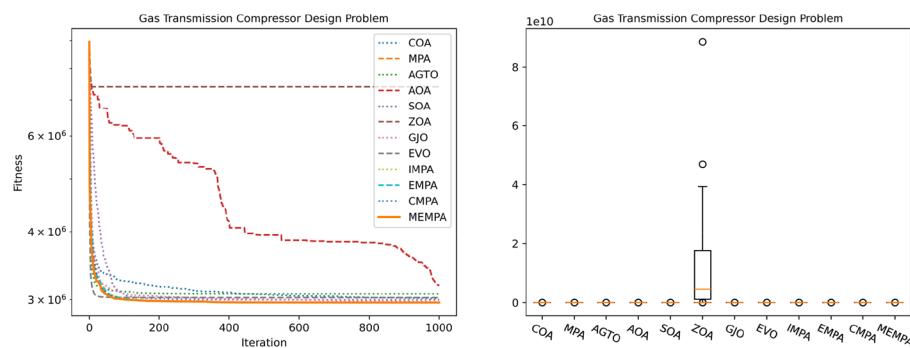


Fig. 20 Convergence curves and boxplots of optimizers in the gas transmission compressor design problem

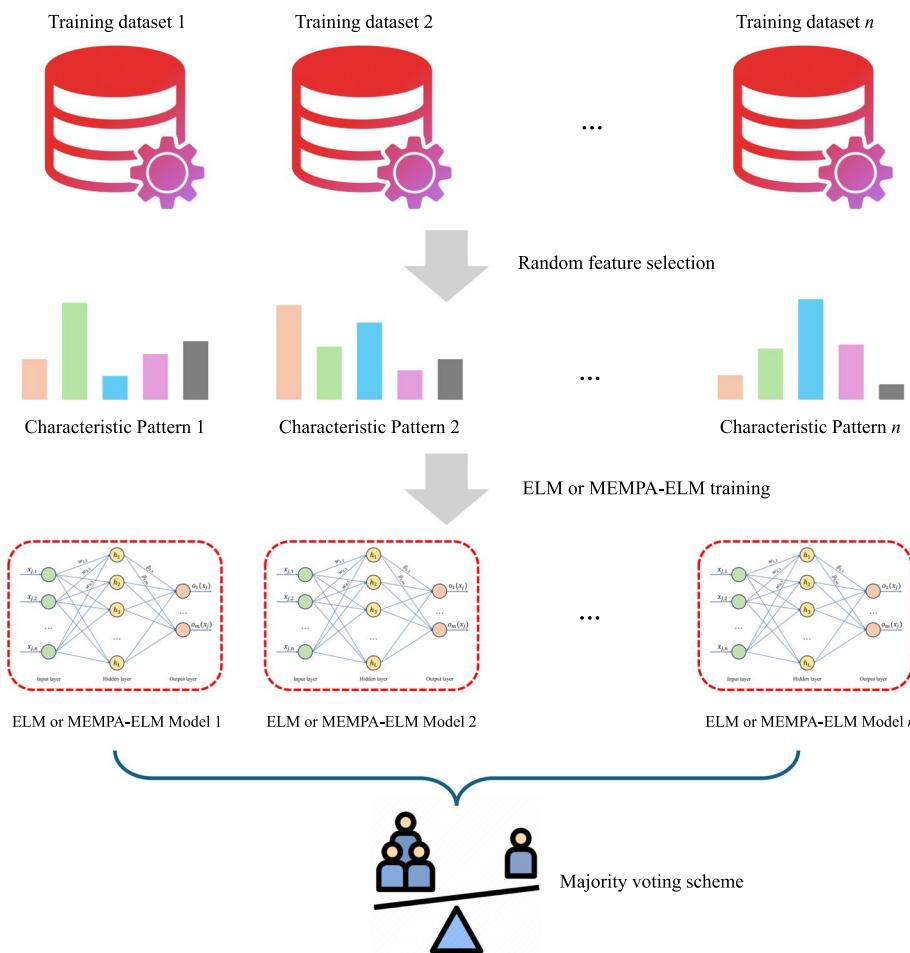
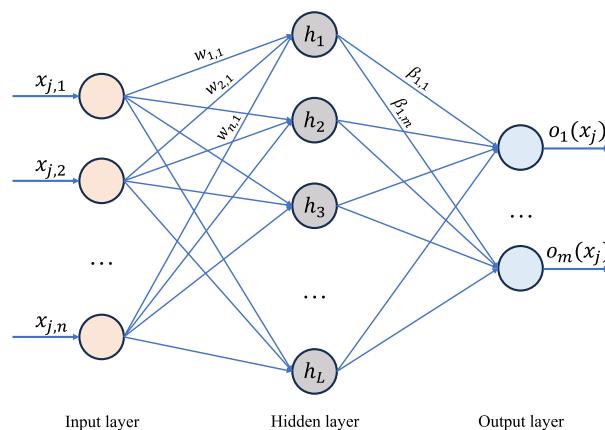


Fig. 21 The framework of the MEMPA-ELM model for intrusion detection

Subsequently, ELM minimizes the mean mean squared error (MSE) as formulated in Eq. (14).

$$\min \| \mathbf{o}_j - \mathbf{t}_j \|^2 \quad (14)$$

**Fig. 22** The architecture of ELM [42]**Table 16** Information of the dataset NSL-KDD

Dataset	Normal	Dos	Probe	R2L	U2R	Total
KDDTrain+	67343	45927	11646	995	52	125973
KDDTest+	9711	7458	2754	2421	200	22544

Subsequently, the output weights β can be determined using Eq. (15).

$$\beta = H^\dagger T \quad (15)$$

where $T = [t_1, t_2, \dots, t_N]^T$ and H^\dagger is the Moore-Penrose generalized inverse of the hidden layer output matrix H .

Objective function

To evaluate the quality of the generated solutions, a fitness function should be defined. This paper utilizes the classification accuracy of the ELM model as the objective function, as defined in Eq. (16).

$$f_{min} = \min(1 - Acc) \quad (16)$$

where the maximization of classification accuracy can be transformed to the minimization of $(1 - Acc)$. Accuracy is the proportion of correctly classified instances out of the total number of instances in the training set, which can be defined in Eq. (17)

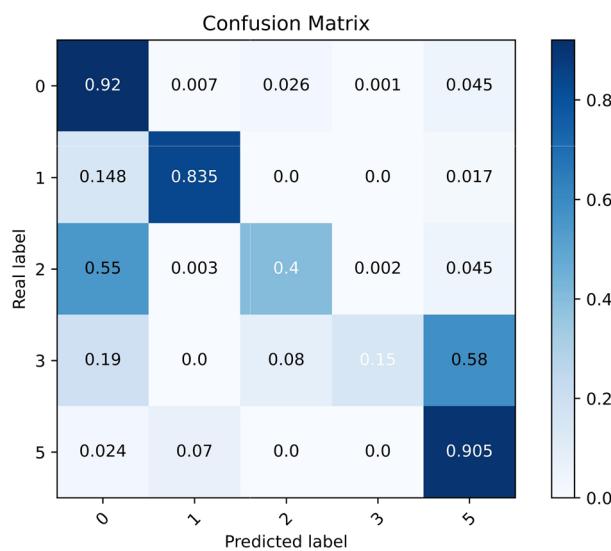
$$Acc = \frac{\sum_{i=1}^n \sum_{j=1}^c f(i,j) C(i,j)}{n} \quad (17)$$

where n is the number of instances in the training set and c is the number of categories. $f(i,j)$ is a function that returns 0 or 1. If instance i belongs to category j , then $f(i,j)$ is 1; If the predicted category of instance i is j , then $C(i,j)$ is 1, otherwise it is 0.

Table 17 Classification accuracy of MAs in the NSL-KDD dataset

	Mean	Std	Best	Worst
COA-ELM	78.494 %	1.145e-02	80.489 %	76.354 %
MPA-ELM	79.107 %	1.510e-02	81.496 %	76.622 %
AGTO-ELM	78.376 %	1.113e-02	80.449 %	75.345 %
AOA-ELM	78.345 %	1.430e-02	81.394 %	75.352 %
SOA-ELM	78.565 %	1.478e-02	81.385 %	75.855 %
ZOA-ELM	78.247 %	1.371e-02	81.274 %	75.669 %
GJO-ELM	78.431 %	1.261e-02	80.369 %	76.689 %
EVO-ELM	78.117 %	1.382e-02	81.039 %	76.645 %
IMPA-ELM	78.917 %	1.302e-02	80.039 %	76.660 %
EMPA-ELM	79.112 %	1.372e-02	81.040 %	76.690 %
CMPA-ELM	79.013 %	1.371e-02	81.029 %	76.385 %
MEMPA-ELM	79.901 %	1.318e-02	82.008 %	78.100 %

Bold indicates best value

**Fig. 23** Optimal confusion matrix of MEMPA-ELM with accuracy=82.008 %

NSL-KDD dataset

The NSL-KDD dataset is an improved version of the original KDD99 dataset collected by MIT Lincoln Laboratory, which consists of four datasets: KDDTest+, KDDtest-21, KDDTrain+, and KDDTrain+20Percent [43]. In addition to normal data, the data set also includes DoS, Probe, U2R, and R2L intrusion data. In this research, KDDTrain+ and KDDTest+ are employed as the training and test datasets, where the distribution is presented in Table 16.

Experiment results

To confirm the effectiveness of MEMPA in intrusion detection, we compare MEMPA-ELM with other MAs. The model of ELM is provided by the hplem library [44] with default parameters, where the population size and maximum iteration of MAs are fixed

at 30 and 20, respectively [45]. The inner parameters of optimizers are consistent with Table 1. All the data are normalized to the range of [0, 1], and the mean, std, best, and worst classification accuracy of MAs are calculated within 30 trial runs. Tables 17 summarize the experimental result, while the optimal confusion matrix of MEMPA-ELM is presented in Fig. 23.

Based on the experimental results and confusion matrix, MEMPA-ELM outperforms the competitor models and achieves state-of-the-art accuracy in intrusion detection. Its superior ability to detect both known and novel threats and enhanced generalization capabilities demonstrate its robustness and reliability across a wide range of challenging scenarios, which highlights significant improvements in precision and false-positive rates.

Conclusion and future works

This paper proposes a Multi-strategy Enhanced Marine Predator algorithm (MEMPA) for numerical optimization. First, the Sobol sequence is employed to generate the initial population and enhance the diversity of the initial population. Second, the mutualism mechanism is introduced to improve the exploitation capacity of MEMPA. Finally, a distance-based selection mechanism is integrated to allow the inferior offspring individual to have a probability of being accepted and surviving, which is expected to improve the ability to avoid local optimality. To evaluate the performance of MEMPA, we conduct numerical experiments in CEC2020 and six classical engineering problems against eleven well-known MAs. By analyzing the mean value, standard deviation, convergence curves, and statistical analysis, the performance of MEMPA is competitive among competitor algorithms. Additionally, we integrate MEMPA with ELM to form MEMPA-ELM to detect the intrusion based on the dataset NSL-KDD, and the average classification accuracy of MEMPA-ELM has achieved 79.9% in the test dataset, which is significantly better than the competitor algorithms.

The above experiments and analysis confirm the advantages of the proposed MEMPA, and we believe that as an efficient optimization tool, MEMPA is promising to handle other optimization challenges such as large-scale global optimization, multi-objective optimization, feature selection tasks, and neural network architecture search tasks.

Author contributions

Zhongmin Wang: Conceptualization, Methodology, Investigation, Writing-original draft, Writing-review & editing, and Funding acquisition. Yujun Zhang: Methodology and Writing-review & editing. Jun Yu: Investigation, Methodology, Formal Analysis, and Writing-review & editing. YuanYuan Gao: Formal Analysis, Validation, and Writing-Review & Editing. Guangwei Zhao: Methodology, Software, and Writing-review & editing. Essam H. Houssein: Investigation, Resources, and Writing-Review & Editing. Rui Zhong: Writing-review & editing, Supervision, and Project administration.

Funding

This work is supported by the project of the School of Tropical Crops, Yunnan Agricultural University (No. 2023RYYB003) and JST SPRING Grant Number JPMJSP2119.

Data availability

The source code of this research can be downloaded at <https://github.com/RuiZhong961230/MEMPA>.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for Publication

Not applicable.

Competing interests

The authors declare no competing interests.

Received: 20 October 2024 Accepted: 17 January 2025

Published online: 19 February 2025

References

1. Rajwar K, Deep K, Das S. An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. *Artif Intell Rev.* 2023;56(11):13187–257.
2. Toaza B, Esztergár-Kiss D. A review of metaheuristic algorithms for solving tsp-based scheduling optimization problems image 1. *Appl Soft Comput.* 2023;148: 110908.
3. Zhong R, Zhang C, Yu J. Cooperative coati optimization algorithm with transfer functions for feature selection and knapsack problems. *Knowl Inf Syst.* 2024;66:6933–74.
4. Zhang Y, Li S, Wang Y, Yan Y, Zhao J, Gao Z. Self-adaptive enhanced learning differential evolution with surprisingly efficient decomposition approach for parameter identification of photovoltaic models. *Energy Convers Manage.* 2024;308: 118387.
5. Agushaka JO, Ezugwu AE, Abualigah L, Alharbi SK, Khalifa HAE-W. Efficient initialization methods for population-based metaheuristic algorithms: a comparative study. *Arch Comput Methods Eng.* 2023;30(3):1727–87.
6. Wang W-C, Tian W-C, Xu D-M, Zang H-F. Arctic puffin optimization: a bio-inspired metaheuristic algorithm for solving engineering design optimization. *Adv Eng Softw.* 2024;195: 103694.
7. Wang W-C, Tian W-C, Hu X-X, Hong Y-H, Chai F-X, Xu D-M. Dttr: Encoding and decoding monthly runoff prediction model based on deep temporal attention convolution and multimodal fusion. *J Hydrol.* 2024;643: 131996.
8. Wolpert DH, Macready WG. No free lunch theorems for optimization. *IEEE Trans Evol Comput.* 1997;1(1):67–82.
9. Zhong R, Yu J, Zhang C, Munetomo M. Srome: a strengthened rime with latin hypercube sampling and embedded distance-based selection for engineering optimization problems. *Neural Comput Appl.* 2024;36(12):6721–40.
10. Barua S, Merabet A. Lévy arithmetic algorithm: an enhanced metaheuristic algorithm and its application to engineering optimization. *Expert Syst Appl.* 2024;241: 122335.
11. Han M, Du Z, Yuen KF, Zhu H, Li Y, Yuan Q. Walrus optimizer: a novel nature-inspired metaheuristic algorithm. *Expert Syst Appl.* 2024;239: 122413.
12. Xu Y, Zhong R, Zhang C, Yu J. Multiplayer battle game-inspired optimizer for complex optimization problems. *Clust Comput.* 2024. <https://doi.org/10.48550/arXiv.2401.00401>.
13. Faramarzi A, Heidarnejad M, Mirjalili S, Gandomi AH. Marine predators algorithm: a nature-inspired metaheuristic. *Expert Syst Appl.* 2020;152: 113377.
14. Aydemir SB. Enhanced marine predator algorithm for global optimization and engineering design problems. *Adv Eng Softw.* 2023;184: 103517.
15. Kumar S, Yıldız BS, Mehta P, Panagant N, Sait SM, Mirjalili S, Yıldız AR. Chaotic marine predators algorithm for global optimization of real-world engineering problems. *Knowl-Based Syst.* 2023;261: 110192.
16. Chen T, Chen Y, He Z, Li E, Zhang C, Huang Y. A novel marine predators algorithm with adaptive update strategy. *J Supercomput.* 2022;79(6):6612–45.
17. Zhang S, Wang S, Dong R, Zhang K, Zhang X. A multi-strategy improved outpost and differential evolution mutation marine predators algorithm for global optimization. *Arab J Sci Eng.* 2023;48:1–24.
18. Han M, Du Z, Zhu H, Li Y, Yuan Q, Zhu H. Golden-sine dynamic marine predator algorithm for addressing engineering design optimization. *Expert Syst Appl.* 2022;210: 118460.
19. Hu G, Zhu X, Wang X, Wei G. Multi-strategy boosted marine predators algorithm for optimizing approximate developable surface. *Knowl-Based Syst.* 2022;254: 109615.
20. Oszust M. Enhanced marine predators algorithm with local escaping operator for global optimization. *Knowl-Based Syst.* 2021;232: 107467.
21. Zhong K, Luo Q, Zhou Y, Jiang M. Tlmpa: Teaching-learning-based marine predators algorithm. *AIMS Mathematics.* 2021;6(2):1395–442.
22. Houssein EH, Hussain K, Abualigah L, Elaziz MA, Alomoush W, Dhiman G, Djenouri Y, Cuevas E. An improved opposition-based marine predators algorithm for global optimization and multilevel thresholding image segmentation. *Knowl-Based Syst.* 2021;229: 107348.
23. Abdel-Salam M, Hu G, Çelik E, Gharehchopogh FS, EL-Hasnony I.M. Chaotic rime optimization algorithm with adaptive mutualism for feature selection problems. *Comput Biol Med.* 2024;179: 108803.
24. Sirsant S, Reddy MJ. Improved mosade algorithm incorporating sobol sequences for multi-objective design of water distribution networks. *Appl Soft Comput.* 2022;120: 108682.
25. Cheng M-Y, Prayogo D. Symbiotic organisms search: a new metaheuristic optimization algorithm. *Comput Struct.* 2014;139:98–112.
26. Ghosh A, Das S, Mallipeddi R, Das AK, Dash SS. A modified differential evolution with distance-based selection for continuous optimization in presence of noise. *IEEE Access.* 2017;5:26944–64.
27. Zhong R, Zhang E, Munetomo M. Cooperative coevolutionary differential evolution with linkage measurement minimization for large-scale optimization problems in noisy environments. *Complex Intell Syst.* 2023;9(4):4439–56.
28. Pierzan J, Dos Santos Coelho L. Coyote optimization algorithm: A new metaheuristic for global optimization problems. In: 2018 IEEE Congress on Evolutionary Computation (CEC), 2018; pp. 1–8.
29. Abdollahzadeh B, Soleimaniyan Gharehchopogh F, Mirjalili S. Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. *Int J Intell Syst.* 2021;36(10):5887–958.

30. Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH. The arithmetic optimization algorithm. *Comput Methods Appl Mech Eng.* 2021;376: 113609.
31. Dehghani M, Trojovský P. Serval optimization algorithm: a new bio-inspired approach for solving optimization problems. *Biomimetics.* 2022;7(4):204.
32. Trojovská E, Dehghani M, Trojovský P. Zebra optimization algorithm: a new bio-inspired optimization algorithm for solving optimization algorithm. *IEEE Access.* 2022;10:49445–73.
33. Chopra N, Mohsin Ansari M. Golden jackal optimization: a novel nature-inspired optimizer for engineering applications. *Expert Syst Appl.* 2022;198: 116924.
34. Azizi M, Aickelin U, Khorshidi H, Baghhalzadeh Shishehgarkhaneh M. Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization. *Sci Rep.* 2023;13:226.
35. Abdel-Basset M, El-Shahat D, Chakrabortty RK, Ryan M. Parameter estimation of photovoltaic models using an improved marine predators algorithm. *Energy Convers Manage.* 2021;227: 113491.
36. Abd Elaziz M, Thanikanti SB, Ibrahim IA, Lu S, Nastasi B, Alotaibi MA, Hossain MA, Yousri D. Enhanced marine predators algorithm for identifying static and dynamic photovoltaic models parameters. *Energy Convers Manage.* 2021;236: 113971.
37. Houssein EH, Saad MR, Hashim FA, Shaban H, Hassaballah M. Lévy flight distribution: a new metaheuristic algorithm for solving engineering optimization problems. *Eng Appl Artif Intell.* 2020;94: 103731.
38. Zhao S, Zhang T, Ma S, Chen M. Dandelion optimizer: a nature-inspired metaheuristic algorithm for engineering applications. *Eng Appl Artif Intell.* 2022;114: 105075.
39. Dehghani M, Montazeri Z, Trojovská E, Trojovský P. Coati optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems. *Knowl-Based Syst.* 2023;259: 110011.
40. Van Thieu N, Mirjalili S. Mealpy: An open-source library for latest meta-heuristic algorithms in python. *J Syst Architect.* 2023;139: 102871.
41. Ahmed M, Naser Mahmood A, Hu J. A survey of network anomaly detection techniques. *J Netw Comput Appl.* 2016;60:19–31.
42. Zhong R, Zhang C, Yu J. Hierarchical rime algorithm with multiple search preferences for extreme learning machine training. *Alex Eng J.* 2025;110:77–98.
43. Su T, Sun H, Zhu J, Wang S, Li Y. Bat: deep learning methods on network intrusion detection using nsl-kdd dataset. *IEEE Access.* 2020;8:29575–85.
44. Akusok A, Björk K-M, Miche Y, Lendasse A. High-performance extreme learning machines: a complete toolbox for big data applications. *IEEE Access.* 2015;3:1011–25.
45. Zhou Y, Zhou N, Gong L, Jiang M. Prediction of photovoltaic power output based on similar day analysis, genetic algorithm and extreme learning machine. *Energy.* 2020;204: 117894.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.